Applying the Wang-Landau Algorithm to Lattice Gauge Theory

Barak Bringoltz and Stephen R. Sharpe

Department of Physics, University of Washington, Seattle, WA 98195-1560, USA

Abstract

We implement the Wang-Landau algorithm in the context of $SU(N)$ lattice gauge theories. We study the quenched, reduced version of the lattice theory and calculate its density of states for $N = 20, 30, 40, 50$. We introduce a variant of the original algorithm in which the weight function used in the update does not asymptote to a fixed function, but rather continues to have small fluctuations which enhance tunneling. We formulate a method to evaluate the errors in the density of states, and use the result to calculate the dependence of the average action density and the specific heat on the 't Hooft coupling λ. This allows us to locate the coupling λ_t at which a strongly first order transition occurs in the system. For $N = 20$ and 30 we compare our results to those obtained using Ferrenberg-Swendsen multi-histogram reweighting and find agreement with errors of 0.2% or less. Extrapolating our results to $N = \infty$ we find $(\lambda_t)^{-1} = 0.3148(2)$. We remark on the significance of this result for the validity of quenched large-N reduction of $SU(N)$ lattice gauge theories.

I. INTRODUCTION

Strongly first order phase transitions provide a difficult challenge for numerical simulations. Consider, for example, a physical system whose interactions are characterized by a single coupling g. A naive approach to estimate the transition coupling, g_t, is to perform Monte-Carlo (MC) simulations at couplings that are close to g_t and locate the point at which different observables are discontinuous. These measurements, however, are affected by a strong hysteresis whose width grows with the number of degrees of freedom $N_{\rm dof}$. For large enough $N_{\rm dof}$, this width dominates the error in the transition coupling, which can result in large uncertainties ($10\% - 20\%$ in the example we consider here).

To obtain improved precision the way forward is undoubtedly to use reweighting algorithms. For example, Ferrenberg-Swendsen reweighting (FSR) uses MC simulations to measure the normalized histogram of the action A for a coupling $g = g_0$ which is close to g_t [1]. By construction, this is given by

$$h_0(A) \sim \rho(A) \times P_{\rm Boltzmann}(g_0; A)\,, \tag{1.1}$$

where $\rho(A)$ is the density of states and $P_{\rm Boltzmann}(g_0; A)$ is the Boltzmann weight. In the $SU(N)$ lattice gauge theories we consider here, g is typically identified with the bare lattice 't Hooft coupling λ, and $P_{\rm Boltzmann}(g_0; A) \sim \exp(A/\lambda)$. A measurement of $h_0(A)$ thus provides an estimate for $\rho(A)$. Using this, one can estimate the histogram $h_\lambda(A)$ at any other coupling by reweighting:

$$h_\lambda(A) \sim h_0(A) \times \exp\left(A/\lambda - A/\lambda_0\right)\,. \tag{1.2}$$

One then determines the coupling λ_t at which the corresponding histogram $h_{\lambda_t}(A)$ takes a double-peak form, as expected for a first-order phase transition. In practice this amounts to calculating the average action $\mathcal{A}$ and its associated specific heat $\mathcal{C}$ as a function of λ

$$\mathcal{A}(\lambda) \;=\; \int\, dA\; h_\lambda(A)\; A, \tag{1.3}$$

$$\mathcal{C}(\lambda) \;=\; \int\, dA\; h_\lambda(A)\; (A - \mathcal{A}(\lambda))^2, \tag{1.4}$$

and finding the coupling λ_t at which $\mathcal{C}(\lambda)$ peaks.

The FSR method has an obvious shortcoming. When reweighting from λ_0 to λ one is "amplifying" the contribution to $\rho(A)$ from field configurations that are important at λ while suppressing those relevant at λ_0. If, however, the field configurations probed at

λ_0 are substantially different from those important at λ, then this amplification can be dominated by statistical noise. This "overlap" problem can cause a large systematic error which may be hard to evaluate. To avoid it one needs to ensure that the field configurations that are important at λ are reasonably sampled when performing measurements at λ_0. In ordinary situations this means that the couplings λ_0 and λ need to be sufficiently close. When $\lambda \simeq \lambda_t$, however, there are field configurations which are very hard to probe. These are the tunneling configurations between the two phases. Thus for reweighting to work in the context of locating a strongly first order phase transition, one requires that a sufficient number of tunneling events are observed whilst measuring the histograms. This requirement can be very restrictive when $N_{\rm dof}$ is large because the tunneling probability typically falls exponentially as $N_{\rm dof}$ increases. Consequently, when performing reweighting, it is crucial to use an algorithm that encourages tunneling events.

In this paper we do not discuss all the different alternatives to FSR (which are discussed, for example, in Ref. [2]).[1] Instead we choose to study (a variant of) the Wang-Landau (WL) reweighting algorithm, which was introduced in the field of statistical mechanics [6], and is particularly well suited for promoting tunneling. In the context of gauge theories, this algorithm can be considered to be a modern incarnation of the early attempts, such as the ones in Ref. [7], to calculate $\rho(A)$ of lattice gauge theories (see also Refs. [8, 9]).

A sketch of the WL algorithm is as follows (a more precise definition will be given in Sec. III). From here on we use the action *density* $E \sim A/N_{\rm dof}$ as our prime observable, and so denote the density of states by $\rho(E)$. We denote the Monte-Carlo time by t and the WL estimate of the density of states at time t by $\rho_t(E)$.

1. Begin at MC-time $t = 0$ with an initial estimate for the density of states, $\rho_0(E)$.

2. Use $1/\rho_t(E)$ as a Boltzmann weight to create a series of field configurations.

3. Update $\rho_t(E) \to \rho_{t+1}(E) = \rho_t(E) + \delta\rho(E)$. The update function $\delta\rho(E)$ depends on the MC history between times t and $t + 1$ in a way that biases *against* small or null

[1] One attractive option is the multi-canonical algorithm of Ref. [3]. We did not use this approach because it had been found in Ref. [4], which studied a model similar to ours, that a very delicate tuning of parameters was needed for large enough systems. One advantage of the WL algorithm is that it is self-tuning. An alternative, applied successfully in Ref. [5], is to use the WL algorithm to provide an estimate of the weight function of the multi-canonical algorithm.

changes of E at time $t+1$, and so encourages tunnelings. This is an essential point in Wang-Landau reweighting (WLR) and we discuss it in greater detail in Section III.

4. GoTo step (2).

One can show, with some assumptions, that for large enough t, $\rho_t(E)$ converges to the vicinity of $\rho(E)$, and subsequently fluctuates around it. We provide this demonstration in Sec. III B, generalizing the discussion in Ref. [10]. The fluctuations are an intrinsic part of the WL algorithm, and are the consequence of the ergodicity enforced by the "biasing" in step (3) above. Once converged, the algorithm generates a chain of field configurations that are weighted by an approximately flat probability function

$$P(E) \sim \rho(E) \times 1/\rho_t(E) \approx E - \text{independent}\,. \tag{1.5}$$

Consequently, all values of E will be accessed with approximately equal probability, including those corresponding to tunneling events. Using the estimate of $\rho(E)$ one can calculate the specific heat $\mathcal{C}(\lambda)$ and locate its peak.

In this paper we adapt the WL algorithm to $SU(N)$ gauge theories and in particular formulate a systematic way to evaluate errors in derived quantities such as $\mathcal{C}(\lambda)$. The model we choose to study is obtained from four-dimensional $SU(N)$ lattice gauge theories by "quenched reduction" to a single lattice site (see, for example, Refs. [11] and the recent review in Ref. [12]). It is a matrix model of four $SU(N)$ matrices. The interactions between these matrices are governed by the 't Hooft coupling λ, and lead to a nontrivial change in various expectation values as one moves from strong to weak couplings. This behavior becomes a strongly first order transition when $N \to \infty$ and it is this transition we wish to analyze using the WL algorithm.

As noted above, we use a variant of the WL algorithm. The key difference between our variant and the original WL algorithm ("WL^0"), is that the latter includes an iterative procedure which we do not use.[2] Namely, in WL^0, the steps (1-4) above are first applied with a given update function, $\delta\rho_1(E)$, for some Monte-Carlo time T_1. The time T_1 is determined "on the fly" by requiring that the values of E that are visited are sufficiently uniform. Once the chosen criterion is fulfilled, the function $\delta\rho_1(E)$ is replaced by $\delta\rho_2(E)$ which is smaller,

[2] A less important difference is that one must adapt the original algorithm from systems with discrete variables to those with continuous degrees of freedom. We describe how this has been done below.

i.e. obeys $|\delta\rho_2(E)| < |\delta\rho_2(E)|$. The procedure is then iterated until the magnitude of $\delta\rho(E)$ drop below the machine accuracy. As shown in [13], and discussed below (for example see Section IV), the tunneling rate, which is what one wishes to increase in the WL algorithm, decreases as the size of $\delta\rho(E)$ is decreased, making the WL^0 less and less efficient as it is iterated. For this reason we keep $\delta\rho(E)$ finite, and thus always have a Boltzman weight which varies (albeit by a small amount) so as to maintain the tunneling rate. This also avoids the need to tune extra parameters, such as the choice of the flatness criterion. Despite the lack of a fixed weight, we can measure expectation values since $\rho_t(E)$ fluctuates around the correct value.

A different solution to the tunneling problem, involving ultimately fixed weights, is presented in Ref. [14].

The outline of the paper is as follows. We first introduce the matrix model that we study in Sec. II. We describe the Wang-Landau algorithm and its properties in Sec. III, and in Sec. IV we describe our implementation and in particular the tuning of parameters and the calculation of errors. In Sec. V we report our results and compare them to corresponding data obtained using Ferrenberg-Swendsen reweighting and standard Monte-Carlo simulations. We summarize in Sec. VI, and remark on the implication of our results to the validity of large-N quenched reduction of $SU(N)$ lattice gauge theories. Appendix A includes a description of the different update algorithms which we use, and Appendix B discusses additional technical issues related to the implementation of the Wang-Landau algorithm.

Our results for the transition coupling λ_t were already quoted in Ref. [12].

II. QUENCHED-REDUCED $SU(N)$ LATTICE GAUGE THEORIES

In this section we briefly describe the matrix model which we study. For a discussion of its relevance to $SU(N)$ gauge theories we refer to Ref. [12] and references therein.

A. Definition of the matrix model

The model consists of four $SU(N)$ matrices $\{V_\mu\,;\,\mu = 1,2,3,4\}$. Observables are built from the $SU(N)$ 'link matrices' U_μ defined by

$$U_\mu \equiv V_\mu \Lambda_\mu V_\mu^\dagger, \tag{2.1}$$

where Λ_μ are the fixed, diagonal $SU(N)$ matrices

$$\Lambda^{ab} \equiv \delta^{ab}\, e^{ip^a_\mu} \;: \qquad p^a_\mu \in [0, 2\pi]\,, \qquad a, b \in [1, N]\,. \tag{2.2}$$

The quenched momenta p^a_μ are drawn from some distribution—various possibilities are discussed in Ref. [12]. Since our focus here is on the algorithm, we pick one choice of momenta (the "clock" momenta),

$$p^a_\mu = \frac{2\pi}{N}\left(a - \frac{N+1}{2}\right) \quad ; \quad a \in [1, N]\,, \tag{2.3}$$

and use it throughout. Expectation values of an observable $\mathcal{O}(\mathcal{U})$ are calculated via

$$\langle \mathcal{O}(\mathcal{U}) \rangle \equiv Z(b)^{-1} \int \prod_\mu DV_\mu \;\exp\left(\, b\, A\right)\, \mathcal{O}(\mathcal{U})\,, \tag{2.4}$$

where here the action A is

$$A = N \sum_{\mu<\nu} 2\mathrm{Re}\,\mathrm{Tr}\,\left(U_\mu U_\nu U^\dagger_\mu U^\dagger_\nu\right)\,, \tag{2.5}$$

and b is the inverse of the 't Hooft coupling, $b = 1/\lambda$. The partition function $Z(b)$ is

$$Z(b) \equiv \int \prod_\mu DV_\mu \,\exp(\, b\, A)\,, \tag{2.6}$$

and DV_μ is the Haar measure on $SU(N)$. The integral over V_μ includes matrices that realize permutations in the indices a of p^a_μ, and so the construction above is invariant under such permutations. Thus one can equally define the model with any set of p^a_μ obtained from Eq. (2.3) by permuting the a indices, independently in each direction.

We take the action density to be

$$E \equiv \frac{A}{12N^2}\,, \tag{2.7}$$

so that it is the average, normalized plaquette. We consider, for simplicity, only even values of N, for which E lies in the range $[-1, 1]$.

B. A sketch of the phase diagram

In Ref. [12] we mapped the phase diagram of the model in b, and saw strong evidence that there exists a first order phase transition at $b = b_t \simeq 0.3$. This was also seen in earlier

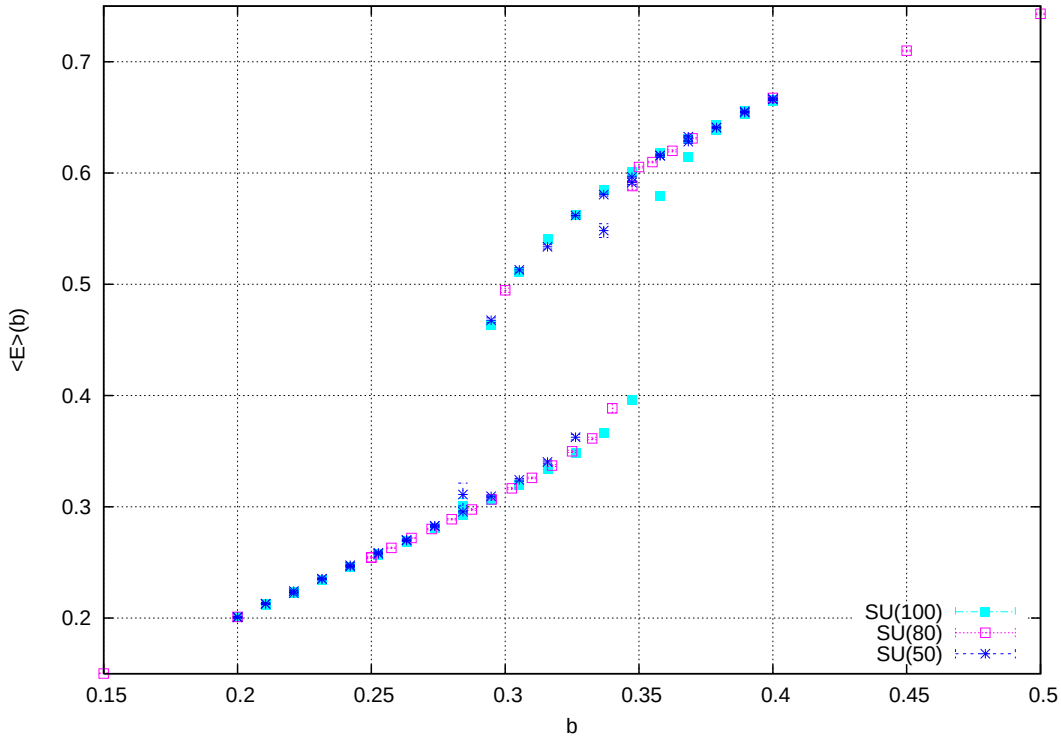

FIG. 1: Hysteresis plots of the average action density E versus $b = 1/\lambda$ for $SU(50)$ ([blue] crosses), $SU(80)$ ([magenta] open squares), and $SU(100)$ ([light blue] filled squares). For details see Ref. [12].

studies of the model (for example in Ref. [15]). To demonstrate this we present in Fig. 1 our results for $\langle E \rangle(b)$, obtained using conventional MC simulations (using algorithms described in Appendix. B). A clear hysteresis is seen, with width increasing with N, as expected since $N_{\rm dof} \propto N^2$. Our aim in this paper is to develop a method that can accurately locate the coupling b_t at which this transition occurs.

III. WANG-LANDAU REWEIGHTING

Reweighting methods start by integrating out all but a few variables (usually one or two) from the partition function. We use a single remaining variable, the action density E:

$$Z(b) = \int DV \, \exp\left(b\, A\right) \equiv \int_{-1}^{1} dE \, \rho(E) \exp\left(12 N^2 \, b \, E\right) \tag{3.1}$$

$$\equiv \int dE \exp\left(\omega(E) + 12 N^2 \, b \, E\right). \tag{3.2}$$

Here $\rho(E)$ is the number density of field configurations with action density in the range $[E, E + dE]$ and $\omega(E)$ is the associated "entropy":

$$\omega(E) = \log\left(\rho(E)\right). \tag{3.3}$$

The expectation value of an observable, $\mathcal{O}(E)$, that depends solely on E can be written as

$$\langle \mathcal{O} \rangle = Z^{-1}(b) \int dE \, \exp\left(\omega(E) + 12\, N^2\, b\, E\right) \mathcal{O}(E). \tag{3.4}$$

Thus, calculating the function $\omega(E)$ can be considered as solving the theory in the sector that couples to operators of the form $\mathcal{O}(E)$.

The original WL algorithm was introduced in Ref. [6] to study statistical systems with discrete degrees of freedom. E then takes discrete values, and this is reflected in the formulation of the original algorithm. In our case, however, E is continuous, and we need to adapt WLR accordingly. Two alternative approaches have been considered in the statistical-mechanics and molecular-dynamics literature: (1) Discretize E into bins and then apply the discrete WL algorithm. This approach has been used, for example, to study the classical Heisenberg model [16]; (2) Generalize the WL algorithm so that it updates $\omega(E)$ treating E as a continuous variable [17, 18].[3] Based on preliminary studies, we chose to pursue only option (2) in detail. We follow and extend the approach suggested in Ref. [18], which we next describe and analyze in some detail.

A. The algorithm

The algorithm proceeds by updating an estimate of the entropy, $\omega_t(E)$, where t is the Monte-Carlo (MC) time. It also updates the histogram of the action, $h_t(E)$, which is an auxiliary quantity used to estimate convergence. The steps of the algorithm are as follows [18]:

1. Make an initial guess for the entropy function at time $t = 0$, $\omega_0(E)$, using any available prior knowledge, such as the results from a related system (e.g. a smaller value of N in our study). Set the histogram to zero: $h_{t=0}(E) = 0$. Pick any starting configuration.

[3] Binning is still required to store functions like $\omega(E)$ in memory, but does not play an essential role in the algorithm. See Appendix B for further discussion.

2. Propose a new field configuration in an unbiased way, and accept it with probability:

$$\mathrm{Prob}(E \to E') = \min\left[\exp\left(\omega_t(E) - \omega_t(E')\right), 1\right], \tag{3.5}$$

where E and E' are respectively the action densities of the original and proposed configurations.

3. Repeat step (2) N_{hit} times for equilibration.

4. Let E_t be the final value of the action density after step (3). Update the entropy as follows:

$$\omega_t(E) \to \omega_{t+1}(E) = \omega_t(E) + \gamma F_\delta(E, E_t), \tag{3.6}$$

where $\gamma > 0$ and F_δ is a fixed, positive function, which smears the update over a range of action density of width $\sim \delta$ centered on E_t, and should be invariant under $E_t \leftrightarrow E$. Possible choices for F_δ are discussed in Ref. [18]—we use a simple Gaussian form

$$F_\delta(E, E_t) = e^{-(E-E_t)^2/\delta^2}. \tag{3.7}$$

5. Update the histogram:

$$h_t(E) \to h_{t+1}(E) = h_t(E) + \delta(E - E_t). \tag{3.8}$$

6. GoTo step (2).

It is important to understand the meaning of the crucial step (4) : if the simulation has spent some time in the vicinity of a particular value of E, then step (4) will increase ω_t in this region, and the update probability (3.5) will favor motion to other regions of E—this is how the WL algorithm encourages tunneling.

As we show in the following subsection, the WL algorithm converges in the sense that, for large enough t, $\omega_t(E) - \omega(E)$ fluctuates around an E-independent constant. This constant drops out when one uses Eq. (3.4) to calculate averages of physical observables, and so it is in this sense that

$$\lim_{t\to\infty} \omega_t(E) = \omega(E) \tag{3.9}$$

in the WL algorithm.

In Section IV, we suggest practical ways to determine how large t needs to be, how to evaluate the errors in the estimate for $\omega(E)$, and how to choose appropriate ranges for the

parameters δ and γ. In our implementation of the algorithm we restrict E to lie in the interval $E_{\min} \leq E \leq E_{\max}$, which is a subset of the full range of values E can take. Apart from the need to begin with a configuration having E inside this range, the only change to the algorithm involve certain boundary effects that we discuss in Appendix B.

B. Theoretical analysis of Wang-Landau reweighting

A theoretical analysis of the original, 'discrete' version, of the WL algorithm, including some of its systematic errors, was given in Ref. [10]. In this section we extend that analysis to the continuous WL algorithm just described.

Consider the probability distribution of the action density E at MC-time t after step (3) of the algorithm has been completed. Assuming that $N_{\rm hit}$ is large enough, this is

$$p_t(E) = \frac{1}{Z_t} \exp\left(\omega(E) - \omega_t(E)\right), \tag{3.10}$$

where the normalization factor Z_t ensures that $\int dE\ p_t(E) = 1$ (where here and in the following the E integral implicitly runs from $E_{\min}$ to $E_{\max}$). This is the probability distribution from which E_t is drawn. After updating $\omega_t(E)$ according to eq. (3.6), the function $p_t(E)$ changes as follows

$$p_t(E) \to p_{t+1}(E) = \frac{1}{Z_{t+1}} \exp\left[\omega(E) - \omega_t(E) - \gamma F_\delta(E, E_t)\right], \tag{3.11}$$

and a simple manipulation gives

$$\frac{p_{t+1}(E)}{p_t(E)} = \frac{\exp\left[-\gamma F_\delta(E, E_t)\right]}{\left\langle \exp\left[-\gamma F_\delta(E, E_t)\right]\right\rangle_t}. \tag{3.12}$$

Here the average $\langle , \rangle_t$ is with respect to the probability distribution at time t,

$$\left\langle f(E)\right\rangle_t \equiv \int dE\, p_t(E)\, f(E). \tag{3.13}$$

In order to understand the convergence properties of the algorithm, we need a measure of the closeness of the estimate $\omega_t(E)$ to the true $\omega(E)$. When $\omega_t(E) = \omega(E)$, the probability distribution (3.10) is flat, i.e. $p_t(E) = p_{\rm flat}(E) = 1/\Delta E$. Thus one possible measure of convergence is

$$\mu_t \equiv \int \frac{dE}{\Delta E} \log\left[\frac{p_t(E)}{p_{\rm flat}(E)}\right] = \int \frac{dE}{\Delta E} \log\left[\Delta E\, p_t(E)\right]. \tag{3.14}$$

This is adapted from the similar discrete quantity used in Ref. [10]. It is straightforward to see that $\mu_t \leq 0$,[4] with the upper bound saturated only when $p_t = p_{\rm flat}$.

We thus consider the change, $\Delta\mu_t = \mu_{t+1} - \mu_t$, between two adjacent time steps:

$$\Delta\mu_t = \int (dE/\Delta E)\, \log[p_{t+1}(E)/p_t(E)] \tag{3.15}$$

$$= \int (dE/\Delta E)\, \{-\gamma F_\delta(E, E_t) - \log\langle \exp\left[-\gamma F_\delta(E, E_t)\right]\rangle_t\}\ . \tag{3.16}$$

Our choice of F_δ, eq. (3.7), satisfies $\int dE\, F_\delta(E, E_t) = \delta\sqrt{\pi}$ for all $E_{\rm min} \leq E_t \leq E_{\rm max}$ (This is true even taking into account any boundary effects – see Appendix B). Thus

$$\Delta\mu_t = -\gamma\delta\sqrt{\pi}/\Delta E - \log\langle \exp\left[-\gamma F_\delta(E, E_t)\right]\rangle_t\, . \tag{3.17}$$

Since $\gamma > 0$, the logarithm is always negative and $\Delta\mu_t$ is bounded from below

$$\Delta\mu_t \geq -\gamma\delta\sqrt{\pi}/\Delta E\, . \tag{3.18}$$

Had this lower bound had been zero, then a monotonic convergence of $\mu_t \to 0$ as $t \to \infty$ would have been possible. A negative lower bound, however, suggests a more complicated behavior involving fluctuations. In the rest of this section we describe the way these fluctuations emerge and quantify how they effect μ_t, $\omega_t(E)$ and $p_t(E)$.

1. *$\Delta\mu_t$ as a function of t and its ensemble average*

We begin by illustrating the possible values that $\Delta\mu_t$ can take. First consider the $\delta \to 0$ limit, in which, assuming also that $\gamma \ll 1$, one finds[5]

$$\Delta\mu_t = \gamma\delta\sqrt{\pi}\,(p_t(E_t) - p_{\rm flat}) + O(\gamma^2)\, . \tag{3.19}$$

Thus if $p_t(E_t)$ is above (below) the flat distribution value $1/\Delta E$, then $\Delta\mu_t$ is positive (negative). For large t, as we will see below, the generic size of $|p_t - p_{\rm flat}|$ is $\sim \sqrt{\gamma}$, so that $\Delta\mu_t$ then scales as $\gamma^{3/2}$.

Second, assume that at time t one has $\omega_t(E) = \omega(E)$ so that $p_t = p_{\rm flat}$ and μ_t takes its maximum value, $\mu_t = 0$. Since the algorithm updates the entropy, $\omega_t(E) \to \omega_{t+1}(E) =$

[4] Given $1 = \int dE\, p_t(E) = \int dE\, e^{log(p_t(E))}$, use the identity $\int \left(\frac{dE}{\Delta E}\right)\, e^{f(E)} \geq \exp\left\{\int \left(\frac{dE}{\Delta E}\right)\, f(E)\right\}$.

[5] The assumption $\gamma \ll 1$ is valid for all our calculations since we use $\gamma \simeq 10^{-4} - 10^{-6}$.

$\omega_t(E) + \gamma F_\delta(E, E_t)$, $\Delta\mu_t$ must be negative. A simple calculation gives

$$\Delta\mu_t = -\frac{\gamma^2\delta\sqrt{\pi}}{\Delta E\,\sqrt{8}}\left(1 - \sqrt{2\pi}\frac{\delta}{\Delta E}\right) + O(\gamma^3)\,. \tag{3.20}$$

The size of this rather special step is parametrically smaller than the generic $O(\gamma^{3/2})$ of the first example. Note that the exact density of states is not a "fixed point" of the algorithm, which may be surprising at first glance, but is in fact an essential feature of the algorithm. It ensures that the simulation explores all values of E in the desired range.

To get a more precise measure of how $\Delta\mu_t$ behaves, we calculate its expectation value averaged over an ensemble of simulations all starting with the same $\omega_t(E)$. The result is

$$\langle\Delta\mu_t\rangle \;\equiv\; \int dE_t\,\Delta\mu_t\,p_t(E_t) \tag{3.21}$$

$$=\; -\frac{\gamma\delta\sqrt{\pi}}{\Delta E} - \int dE_t\;p_t(E_t)\,\log\left[1 - \langle 1 - \exp\left(-\gamma F_\delta(E, E_t)\right)\rangle_t\right]. \tag{3.22}$$

(Here the internal average is over E.) Using the identity $\log(1-x)^{-1} > x$ we find

$$\langle\Delta\mu_t\rangle > -\frac{\gamma\delta\sqrt{\pi}}{\Delta E} + \int dE_1\,dE_2\;p_t(E_1)\;p_t(E_2)\;\left\{1 - \exp\left[-\gamma F_\delta(E_1, E_2)\right]\right\}\,. \tag{3.23}$$

The kernel in the curly braces is positive semi-definite, but decreases rapidly towards zero for $|E_1 - E_2| \gg \delta$. It is also symmetric under $E_1 \leftrightarrow E_2$. Consequently, the double integral on the r.h.s. of (3.23) provides a definition of a (smeared) inner product of $p_t(E)$ with itself.

2. *Relation of $\Delta\mu_t$ to $p_t(E)$*

To make use of Eq. (3.23) we must evaluate the second term on the r.h.s. and relate it back to μ_t. For that purpose we first use the kernel and define the following squared "distance" between two probability distributions (3.23):

$$||p_a - p_b||^2 \equiv \frac{\int dE_1\,dE_2\,\left[p_a(E_1) - p_b(E_1)\right]\,\left\{1 - \exp\left[-\gamma F_\delta(E_1, E_2)\right]\right\}\left[p_a(E_2) - p_b(E_2)\right]}{\int (dE_1/\Delta E)\,(dE_2/\Delta E)\,\left\{1 - \exp\left[-\gamma F_\delta(E_1, E_2)\right]\right\}}\,. \tag{3.24}$$

The normalization is chosen so that $||p_{\rm flat}||^2 = 1$. Equation (3.24) is a generalization of the standard Euclidean distance used in [10]. In fact, if one takes $\delta \to 0$, the kernel becomes proportional to $\delta(E_1 - E_2)$, and one obtains (the continuous E version of) the Euclidean distance:

$$\lim_{\delta\to 0}||p_a - p_b||^2 = \int\left(\frac{dE}{\Delta E}\right)\left(\frac{p_a(E) - p_b(E)}{p_{\rm flat}}\right)^2\,. \tag{3.25}$$

For $\delta > 0$, the kernel gives different weights to different Fourier components (in E-space) of $(p_a(E) - p_b(E))$: wavelengths larger than δ are included with full weight, with the weight decreasing to zero as the wavelength itself decreases to zero. As a result UV differences are filtered out. Indeed this kernel is a natural integration measure for our purposes because the WL algorithm only makes changes to ω_t which have wavelengths of $O(\delta)$ or longer.

We can use the distance $||p_t - p_{\rm flat}||$ as an measure of the approach of p_t to $p_{\rm flat}$. To evaluate this distance we need to calculate the integral in the denominator of Eq. (3.24)

$$\int (dE_2/\Delta E)\,\{1 - \exp\left[-\gamma F_\delta(E_1, E_2)\right]\} \equiv [1-c]\gamma\delta\sqrt{\pi}/\Delta E\,. \tag{3.26}$$

The constant c obeys $0 < c \leq 1$, and for small γ is

$$c = \frac{\gamma}{\sqrt{8}} + O(\gamma^2)\,. \tag{3.27}$$

It is independent of E_1 up to boundary effects of $O(\gamma^2\delta/\Delta E)$. Ignoring these numerically very small effects, it is straightforward to show that

$$||p_t - p_{\rm flat}||^2 = ||p_t||^2 - 1\,. \tag{3.28}$$

Combining Eqs. (3.23), (3.28) and (3.26), we find

$$\langle \Delta\mu_t \rangle > \frac{\gamma\delta\sqrt{\pi}(1-c)}{\Delta E}\,\left(||p_t - p_{\rm flat}||^2 - R^2\right). \tag{3.29}$$

with $R^2 \equiv c/(1-c) \simeq \gamma/\sqrt{8}$.

From this it follows that

- If $||p_t - p_{\rm flat}|| > R$ then $\langle \Delta\mu_t \rangle > 0$ and the simulation will, on average, move towards the desired point $p_t = p_{\rm flat}$, at which $\omega_t(E) = \omega(E)$.

- If $||p_t - p_{\rm flat}|| < R$ then the lower bound on $\langle \Delta\mu_t \rangle$ is negative and the simulation can move both towards and away from $p_t = p_{\rm flat}$.

Finally, note that when $|(p_t - p_{\rm flat})/p_{\rm flat}| \ll 1$, one can show from the definition of μ_t [Eq. (3.14)] that

$$\mu_t \approx -\frac{1}{2}\int \left(\frac{dE}{\Delta E}\right)\,\left(\frac{p_t(E) - p_{\rm flat}(E)}{p_{\rm flat}}\right)^2 \approx -\frac{1}{2}||p_t - p_{\rm flat}||^2\,. \tag{3.30}$$

The first approximate equality assumes that the fluctuations of $(p_t(E) - p_{\rm flat})$ are small (which is a good approximation at large t, as we will see shortly). The second approximate equality assumes that the corrections to the $\delta \to 0$ limit, Eq. (3.25), are small, and is thus only an order of magnitude approximation.

Putting together the above ingredients, the following picture emerges. Consider the infinite dimensional space of probability distributions $p_t(E)$, with distances defined by the Euclidean metric Eq. (3.24). Let the origin be at $p_t(E) = p_{\rm flat}$, and denote the radial coordinate in this space, $||p_t - p_{\rm flat}||$, by r. A crucial role is then played by a ball of radius $R \approx \sqrt{\gamma/\sqrt{8}}$ centered at the origin. The results above imply that if $p_t(E)$ lies outside this ball, then the simulation will perform a directed random walk towards the ball, with steps in μ_t [and thus, from Eq. (3.30), also in r^2] of average size proportional to $(r^2 - R^2) \times \gamma\delta/\Delta E$. Since the steps get, on average, smaller as one approaches the ball, the approach to its surface is exponentially slowed. Individual steps, however, do not shrink to zero, so one will eventually end up inside the ball. Once inside, Eq. (3.29) only gives a lower bound on $\langle \Delta\mu_t \rangle$, so we do not know its sign. The simulation may move throughout the ball, or it may cluster near the surface. Combining Eqs. (3.30) and (3.20), one finds that the typical step size is $\Delta r \approx \gamma\delta$ and thus much smaller than the size of the ball $R \sim \gamma^{1/2}$. One possible behavior is illustrated in Fig. 2.

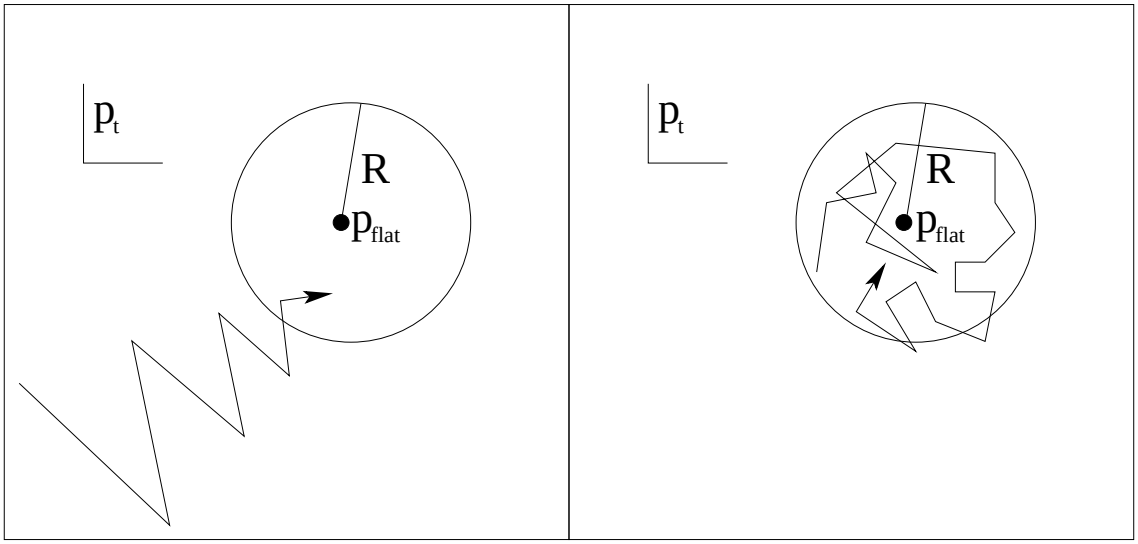

FIG. 2: Pictorial representation of how the Monte-Carlo time history of the WL algorithm looks in $p_t(E)$ space (see text). Left panel: Initial stage - convergence of $p_t(E)$ towards a 'ball' of radius R around $p_{\rm flat}$. The step-size gets smaller as the algorithm approaches the ball. Right panel: Second stage - fluctuations within the ball. If the fluctuations drive p_t outside of the ball, it is driven back inside by the type of motion in the left panel.

It is clear from the foregoing that only in the second stage, when the simulation has reached the ball, can one use $\omega_t(E)$ as an estimate of $\omega(E)$. We now calculate the size of the fluctuations in this estimate. This requires that we remove the overall uniform growth of ω_t which occurs because of the addition of $\gamma F_\delta(E, E_t)$ with a uniform distribution of E_t. To do so we write:

$$\omega_t(E) - \omega(E) = C(t) + \Delta\omega_t(E)\,, \qquad \text{with} \qquad \int dE\,\Delta\omega_t(E) = 0\,. \tag{3.31}$$

The E-independent quantity $C(t)$ is determined by the normalization condition on $\Delta\omega_t$, and is a linear function of t with slope $\delta\gamma\sqrt{\pi}/\Delta E$. $\Delta\omega_t(E)$ contains the physically relevant fluctuations, since $C(t)$ makes no contribution to observables.

Once we are inside the ball we have $||p_t - p_{\rm flat}|| \lesssim R \approx \sqrt{\gamma/\sqrt{8}} \ll 1$. Our task is to use the definition of p_t in Eq. (3.10) to convert this into a result for the fluctuations in ω_t. To do so, we assume that once in the ball, the proximity of $p_t(E)$ to $p_{\rm flat}(E)$ occurs not just on average (as the smallness of $||p_t - p_{\rm flat}||$ implies) but also for each E separately. Then we have that

$$\frac{p_t(E) - p_{\rm flat}}{p_{\rm flat}} \approx \log\left(\frac{p_t(E)}{p_{\rm flat}}\right) = -\Delta\omega_t(E) + O\left(\Delta\omega_t(E)\right)^2\,. \tag{3.32}$$

The last step follows by expanding Z_t in $\Delta\omega_t(E)$. Inserting the result (3.32) into Eq. (3.24) we find the desired relation:

$$||p_{\rm flat} - p_t||^2 \approx \frac{\int dE_1 dE_2 (\omega_t(E_1) - \omega(E_1))\left(1 - \exp\left[-\gamma F_\delta(E_1, E_2)\right]\right)(\omega_t(E_2) - \omega(E_2))}{\int dE_1 dE_2\left(1 - \exp\left[-\gamma F_\delta(E_1, E_2)\right]\right)} \equiv (\Delta\omega)^2\,. \tag{3.33}$$

Thus, once in the ball, the fluctuations in ω_t are the same as those in p_t. Such a "filtered" measure of fluctuations is sufficient, because the update of ω_t does not introduce UV noise. We conclude that $\Delta\omega \approx R \approx \sqrt{\gamma/\sqrt{8}}$. This is the same parametric behavior as in the discrete WL algorithm [10].

An important issue for the practical application of our variant of the WL algorithm is the detailed nature of the fluctuations $\omega(E) - \omega_t(E)$. In particular, do they average to zero *for each* E once one is inside the ball? The previous analysis does not directly address this question. We consider it very plausible, however, that the answer is positive. This is because the algorithm is designed to smooth out nonuniformities in $\omega(E) - \omega_t(E)$, although it does so with some "overshoot" which leads to the fluctuations. It would be interesting to extend the analysis of the algorithm to include such non-equilibrium effects. For the present, however, we assume that $\omega(E) - \omega_t(E)$ fluctuates symmetrically about zero for each E.

We end this subsection by estimating the parametric dependence of fluctuations in the histogram $h_t(E)$, at least in certain limits. The entropy is related to the histogram by a Gaussian transform,

$$\omega_t(E) - \omega_0(E) = \gamma \int dE'\, h_t(E')\, e^{-(E-E')^2/\delta^2}\,, \tag{3.34}$$

where ω_0 is the initial guess. We will use a shorthand notation for this transform and its inverse:

$$(\omega_t - \omega_0) = \gamma \mathcal{G}(h_t)\,, \qquad \mathcal{G}^{-1}(\omega_t - \omega_0) = \gamma\, h_t. \tag{3.35}$$

Using Eq. (3.31), we can write

$$\omega_t(E) - \omega_0(E) = C(t) \,+\, (\omega(E) - \omega_0(E)) + \sqrt{\gamma} f_t(E)\,. \tag{3.36}$$

For large t, when p_t is in the ball, $f_t = \Delta\omega_t/\sqrt{\gamma}$ fluctuates around zero with an amplitude of $O(1)$. Substituting Eq. (3.36) into Eq. (3.35) we obtain

$$h_t = \frac{C(t)}{\gamma\delta\sqrt{\pi}} \,+\, \mathcal{G}^{-1}\left(\omega - \omega_0\right) \,/\, \gamma + \mathcal{G}^{-1}(f_t)/\sqrt{\gamma}\,, \tag{3.37}$$

where we use that result that an inverse Gaussian transform of a constant is a constant.

The subsequent analysis depends on the relative size of the second and third terms in (3.37), and thus on the accuracy of the initial guess $\omega_0(E)$. One extreme case is a poor guess, $\omega_0 = 0$. In this case the second term dominates over the third (at least for small enough γ) which means that if one evaluates the variance in h_t,

$$\delta h_t \;\equiv\; \sqrt{\int (dE/\Delta E)\left(h_t(E) - \overline{h_t}\right)^2}\,, \tag{3.38}$$

$$\overline{h_t} \;\equiv\; \int (dE/\Delta E)\, h_t(E). \tag{3.39}$$

it will have a t-independent contribution proportional to $1/\gamma$.

The other extreme is when one starts with a very good guess, $\omega_0(E) \approx \omega(E)$, so that the fluctuation term in (3.36) dominates over the second term on the r.h.s.. If so, then we expect a t-dependent contribution to δh_t that scales with $1/\sqrt{\gamma}$. Presumably if the second and third terms compete, the scaling will lie somewhere between these two limiting cases. This appears to be the situation in many of our simulations.

The result (3.34) tells us nothing, however, about the UV fluctuations in h_t, since these are filtered out by the Gaussian transform. As discussed further below, we expect that this UV noise increases with t. In practice it is a small contribution in our simulations.

C. The effect of a non-equilibrated Wang-Landau simulation

We close this section by stressing that the analysis just presented is predicated on letting the simulation equilibrate after an update to ω_t is performed. While this equilibration is guaranteed if we let $N_{\rm hit} \to \infty$, most of our runs were done with $N_{\rm hit} = 1$. This means that the analysis above does not directly apply to such simulations—$p_t(E)$ is changing after each update,

$$p_t(E) \to p_{t+1}(E) = p_t(E) + O(\gamma), \tag{3.40}$$

so exact equilibration cannot occur. Nonetheless, when $\gamma \ll 1$ and $p_t(E) \simeq p_{t+1}(E)$, approximate equilibration is possible. Thus we think it is plausible that the analysis just given remains applicable given γ is small enough. We have checked this in practice by doing runs with $N_{\rm hit} \gg 1$ and seeing that the results are unchanged within errors. An example is shown below.

IV. IMPLEMENTING AND TUNING THE WANG-LANDAU ALGORITHM

In this section we describe how we implement the WL algorithm in practice, how we use it to estimate $\omega(E)$ and derived quantities, and suggest criteria for tuning γ, δ and $N_{\rm hit}$.

That tuning is necessary is apparent from the analysis of the previous Section. Particularly crucial is the tuning of γ, which involves a balance between two competing effects. On the one hand, γ controls the speed with which the algorithm explores values of action density. If the simulation has spent some time in the vicinity of a particular value of E, then ω_t will be increased in this region, and the update probability (3.5) will favor motion to other regions of E. The rate of build-up of ω_t is proportional to γ, so the rate of motion through "E-space" will increase with increasing γ. On the other hand, by reducing γ one reduces the fluctuations in ω_t (since $\Delta\omega \propto \sqrt{\gamma}$), and correspondingly reduces statistical errors in quantities derived from ω.

A. Algorithm structure and tuning γ

For our application we can restrict the range of E to $[E_{\rm min}, E_{\rm max}] \subset [-1, 1]$, since we are only interested in the transition region. The range should be large enough that the

errors in the quantities of interest due to this truncation are much smaller than those from statistics. The appropriate range in our case can be read off from the hysteresis curves of Fig. 1. One must cover the transition region and add a conservative cushion on each side, and we typically use $E \in [0.1, 0.7]$ (see below for all our parameter choices). Working with less than a third of the full range $[-1, 1]$ saves considerable computation time.

Having made the choice of range, the algorithm proceeds in two stages.

1. Initial stage:

During this stage the simulation makes a directed random walk towards the "ball" in p_t space of radius R centered on the desired flat distribution. The algorithm explores the chosen range of E and transforms the starting guess $\omega_0(E)$ into a reliable estimate of the actual entropy.

The histogram $h_t(E)$ is a useful monitor of progress during this stage. At the beginning, it will build up non-uniformly because the guess for the entropy function is imperfect, but by the end the histogram should be growing uniformly in E. The variance of the histogram, δh_t, will grow from its initial value of zero and then approximately saturate. This saturation marks the end of the initial stage.

As discussed in the previous section, the value at which it saturates depends on the accuracy of the initial guess. We illustrate in Fig. 3 what happens with both a poor and a good guess. In the former case (data represented by [red] pluses) we start with no information on the entropy, i.e. $\omega_0 = 0$. The histogram first increases for small E, where $\omega(E)$ is large. When $\omega_t(E) \simeq \omega(E)$ for these values of E, the WL random walk gradually starts exploring larger values of E which have lower $\omega(E)$. Eventually (not shown) the whole range is covered, and the histogram grows uniformly, while maintaining in its shape the "memory" of the initial ω_0. This shape is the second term in Eq. (3.37).

The case of a good guess is shown by the [green] crosses. Here we show only the histograms starting after 55×10^5 updates, so as to avoid cluttering the figure. Earlier histograms are similarly horizontal. For these simulations the third (fluctuation) term in Eq. (3.37) may dominate over the second.

The behavior of δh_t for the case of the poor guess is shown in Fig. 4. The stage of rapid

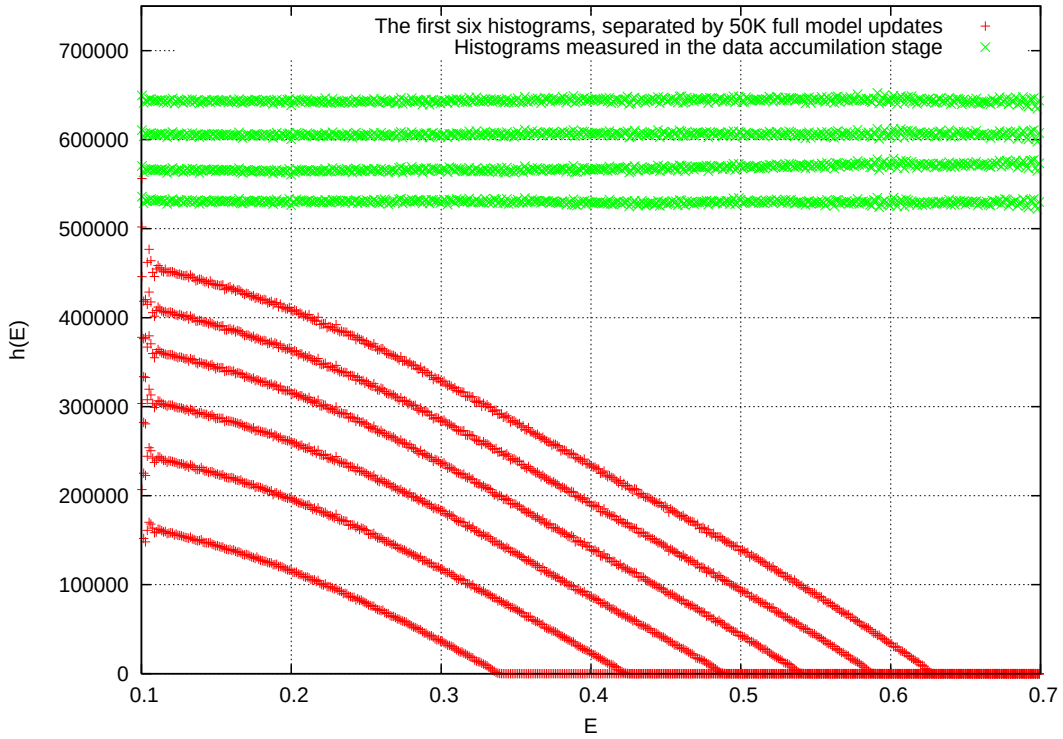

FIG. 3: Comparison of histograms $h_t(E)$ in the initial stage with $\omega_0 = 0$ (red pluses) and the data accumulation stage for a good initial guess for ω_0 (green crosses). Results using WLR for $SU(20)$ with $\gamma = 10^{-4}$, $\delta = 0.005$. The values of t that correspond to the bottom six histograms $t = [5, 10, 15, 20, 25, 30] \times 10^4$ full updates of the model (corresponding to values of 25–150 on the "MC-time" axis of the next figure). The top four histograms were obtained after $[50, 55, 60, 65] \times 10^4$ full updates. The definition of a "full update" in given in Appendix A.

growth ends when δh_t saturates to a nearly constant function of t.[6] There is a small but noticeable residual growth in δh_t which is due, we think, to UV fluctuations in the histogram. As discussed in Sec. III B, these fluctuations are not suppressed by the WL algorithm, and we expect them to be Gaussian with a contribution to δh_t growing like $\sqrt{h_t}$.

Once one has obtained a good estimate of $\omega(E)$ for one value of N, one can scale it with $N_{\rm dof} \propto N^2$ to use as a guess for a different value of N, or reuse it for the same N with a different γ, δ etc.. With a good guess the initial stage is shorter,[7] and, according

[6] As noted in the previous section, we expect the amplitude at saturation to scale with $1/\gamma$, although we have not checked this in this case.

[7] Ref. [18] reports that introducing an update to $\omega_t(E)$ which is applied simultaneously to all values of E can also reduce the computational cost of this initial stage by an order of magnitude. We did not test this extensively.

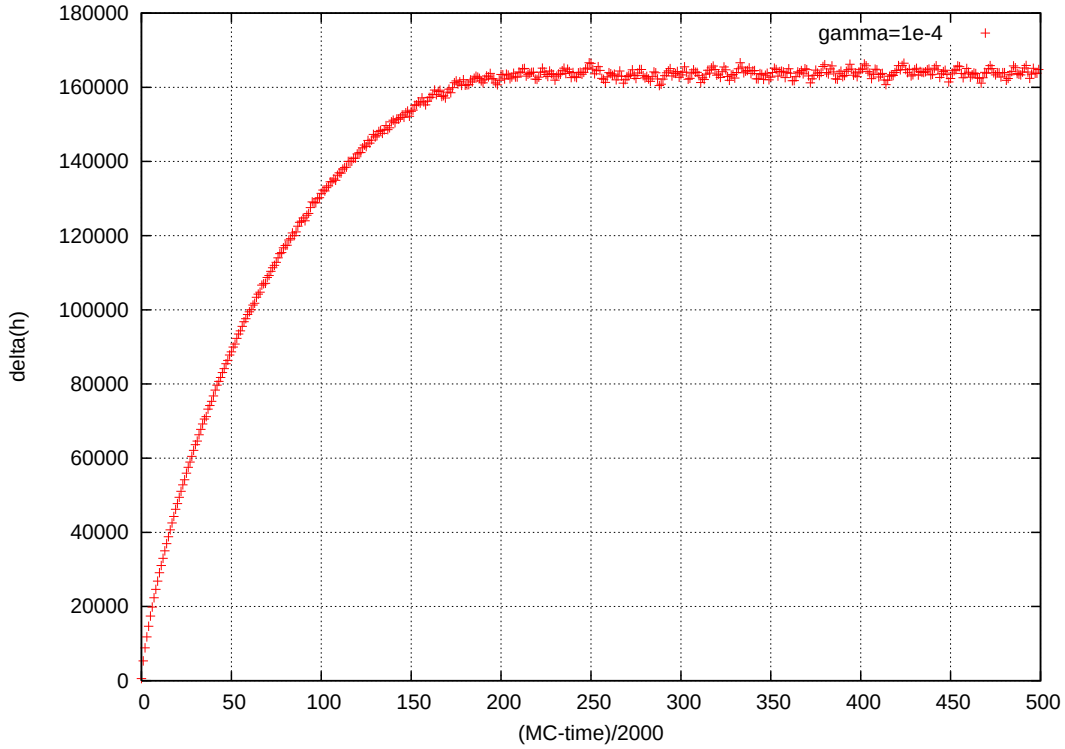

FIG. 4: The average histogram variance, δh_t, for the $\omega_0 = 0$ data presented in Fig. 3.

to Section III B 3, the value of δh_t at saturation should scale more like $1/\sqrt{\gamma}$ than like $1/\gamma$. An example of this situation is shown in Fig. 5 where it appears that most, if not all, of the data is in the "saturation regime", although it is hard to pinpoint exactly the beginning of this regime because of the fluctuations. Note that the $\gamma = 10^{-4}$ data correspond to the "good guess" histograms in Fig. 3 above. The figure shows clearly that the saturated δh_t grows with decreasing γ. The saturated values are approximately 5×10^{-4}, 8×10^{-3} and 2.5×10^{-3}, which are roughly consistent with the expected γ-scaling.

Finally we note that we found it useful to experiment during this initial stage with values for γ, and determine a lower bound such that the range $[E_{\min}, E_{\max}]$ can be explored repeatedly with the available computational resources.

2. Data accumulation stage:

The simulation is now fluctuating around the actual $\omega(E)$ (it is in the ball – see Section III B 3). We propose that one perform N_{meas} measurements of $\omega_t(E)$ separated by a fixed number updates. In our case of a first-order transition, the gap between measurements

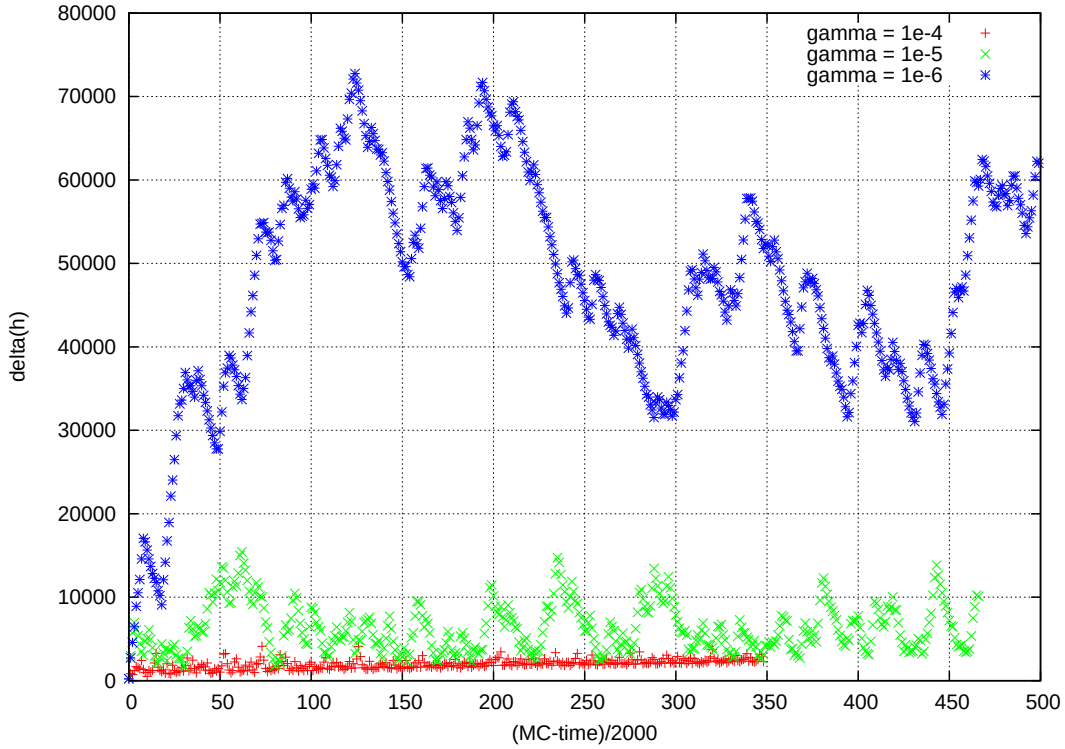

FIG. 5: The standard deviation in the average histogram, δh_t, versus MC-time, for $SU(20)$ and $\gamma = 10^{-4}, 10^{-5}, 10^{-6}$ (shown by [red] pluses, [green] crosses and [blue] stars, respectively). This data uses a good initial guess $\omega_0(E)$ and so has a very short initial stage. Thus δh_t is mostly or completely saturated. Results are for $\delta = 0.005$ and $N_{\rm hit} = 1$. The corresponding plot for $N_{\rm hit} = 20$ and $\gamma = 10^{-5}$ is similar.

should ideally include, on average, several tunneling events.[8] The average of these measurements provides an estimate for $\omega(E)$ (up to an overall irrelevant constant). The deviation of this estimate from the true entropy will then scale as $\sqrt{\gamma/N_{\rm meas}}$.

For derived quantities such as the specific heat (1.4), we propose calculating the errors using the jack-knife or similar method applied to the set of $N_{\rm meas}$ measurements of $\omega_t(E)$. This has the advantage of automatically taking into account correlations in cases where we have two few tunneling events between measurements. We expect the errors in derived quantities to also scale as $\sqrt{\gamma/N_{\rm meas}}$.

We show an example of the behavior of E during this data accumulation stage in Fig. 6. The runs are the same as for Fig. 5, except that we show only the smallest and largest values

[8] We define a tunneling event as motion from $E_{\rm min}$ to $E_{\rm max}$ and back again. This is a conservative definition since a tunneling can occur without motion all the way to the edges of the range.

of γ for the sake of clarity. Tunneling is clearly seen,[9] with a frequency that decreases with

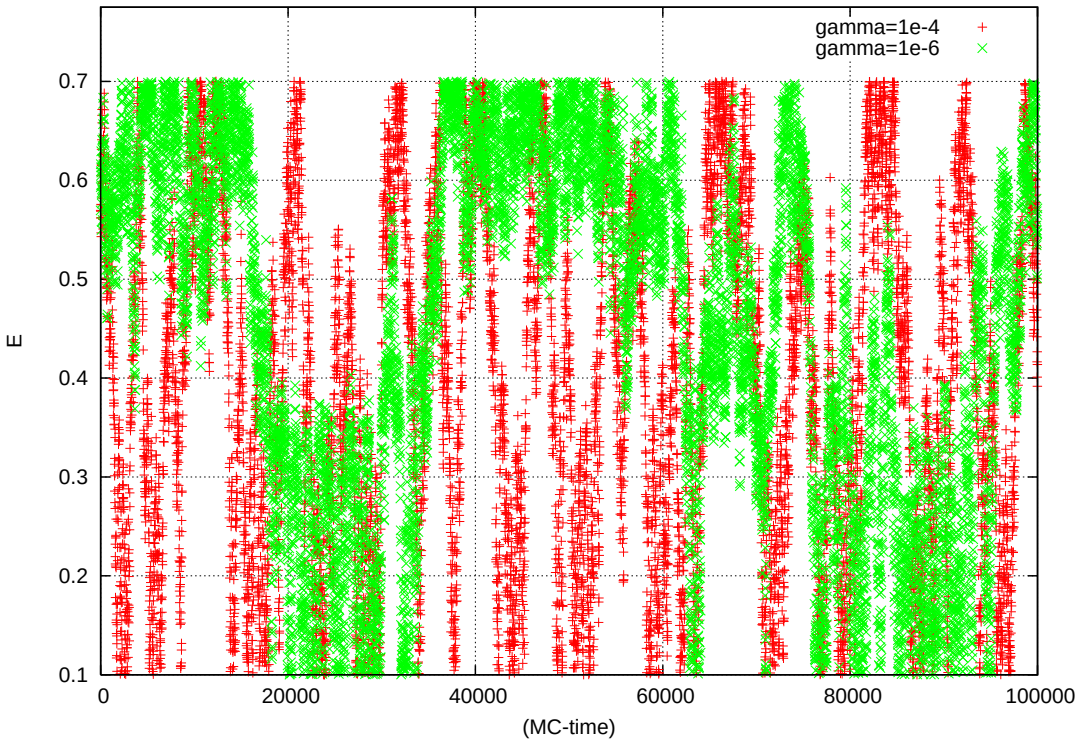

FIG. 6: The MC-time history of the action density E for $SU(20)$ with $\gamma = 10^{-4}$ (red pluses) and $\gamma = 10^{-6}$ (green crosses). In both cases $\delta = 0.005$ and $N_{\rm hit} = 1$.

decreasing γ.

This time history allows one to understand the large fluctuations in δh_t seen for small γ in Fig. 5. Before a tunneling event takes place, the histogram grows only for low (high) values of E, and consequently δh_t grows. After the tunneling, the previously unvisited high (low) range of E is explored and δh_t drops. Thus a tunneling event is manifest in the MC-time history of δh_t as a peak. Indeed we have confirmed that the peaks in Fig. 5. peaks coincide with tunneling events seen in the time histories of E shown in Fig. 6.

The data accumulation stage can also be used to further tune the value of γ. Decreasing γ reduces errors, but also, for fixed computer time, decreases tunneling rates and thus $N_{\rm meas}$. One should choose γ to optimize the error in the derived quantities of most interest, which,

[9] It is important to keep in mind that these tunneling histories inevitably look different from those in canonical simulations running at or near the transition coupling. In the latter the fluctuations in each phase are over a very limited range of E, while in the WL algorithm the simulation must, by construction, move out to the boundaries of the E-range so that all values of E are equally populated.

as we have noted, are expected to scale like $\sqrt{\gamma/N_{\rm meas}}$.

One must also determine how to scale an optimized γ between different values of N. One criterion is to maintain the same tunneling rate. Since $\omega(E)$ is an extensive quantity scaling like $N_{\rm dof}$, we expect that γ must be scaled similarly if it is to lead to a similar rate of motion through E-space, and in particular to the same tunneling rate. We have found that such a scaling rule works reasonably well in practice. As an example, we show in Fig. 7 the comparison of the time-history for $SU(20)$ with $\gamma = 10^{-5}$ and $SU(50)$ with $\gamma = 3 \times 10^{-5}$. The ratio of the γ's is 3 while the ratio in the number of degrees of freedom is $(50/20)^2 \simeq 6$.

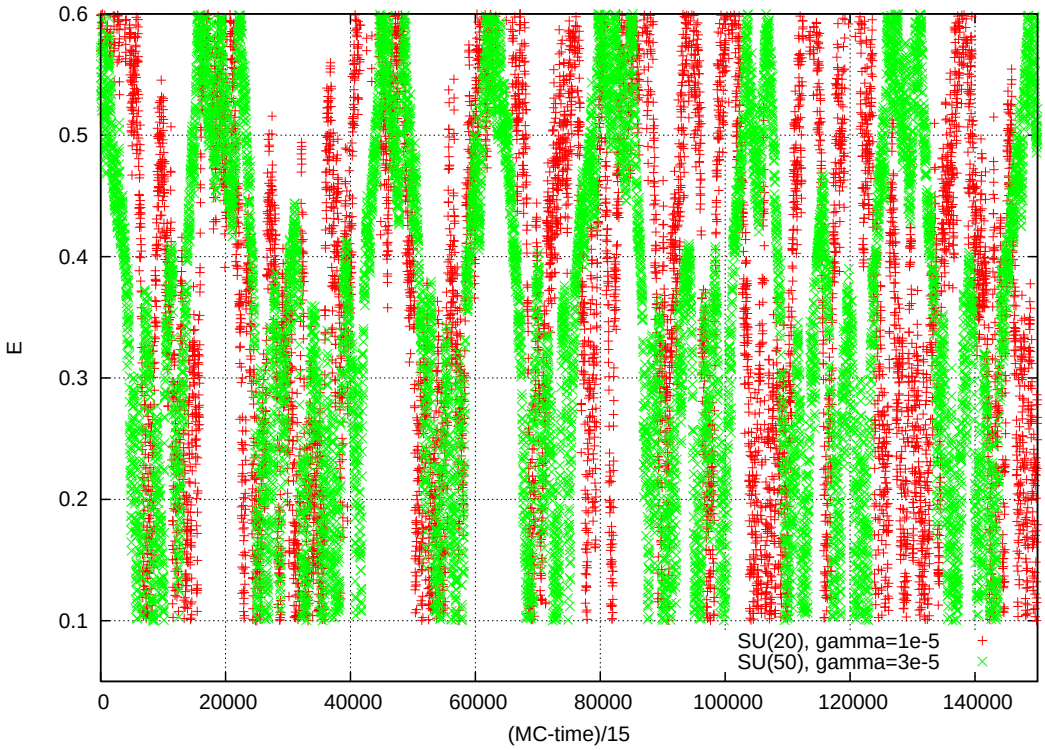

FIG. 7: The MC-time history of the action density E using the WL algorithm. Red pluses: $SU(20)$ with $\gamma = 10^{-5}$. Green crosses: $SU(50)$ with $\gamma = 3 \times 10^{-5}$. In both cases $\delta = 0.005$ and $N_{\rm hit=1}$.

Thus they are of the same order of magnitude and we expect the tunneling rate for $SU(50)$ to be similar to that for $SU(20)$. As the Figure shows this is approximately true. This should be contrasted with standard MC simulations in which the tunneling rate for $SU(50)$ is exponentially smaller, reduced in the present case by about a factor of 500 compared to $SU(20)$. This is a striking example of the efficacy of the WL algorithm at overcoming the suppression of tunneling events.

B. Tuning δ

The parameter δ determines the width of the smearing function $\gamma F_\delta(E, E_t)$ that is added to $\omega_t(E)$. Since the area under γF_δ is proportional to $\gamma \times \delta$, it is this product that determines how fast the simulation moves through E-space. Indeed this product enters in the bound on the steps in $\Delta\mu_t$, Eq. (3.29). By contrast, the size of fluctuations in ω_t, and thus in derived quantities, depends only on γ and not on δ (since $R^2 \approx \gamma/\sqrt{8}$). In light of this one wants to make δ as large as possible before tuning γ.

The upper limit on δ is set by different considerations. As δ increases, the resolution with which one obtains $\omega(E)$ is decreased, and it must not approach the width of the region which makes the important contributions to observables like the specific heat. Thus we propose that one must keep $\delta \ll \sigma$, with σ the width in E of each branch of the canonical distribution $P_C(E)$,

$$P_C(E) \sim \exp\left(\omega(E) + 12N^2 b_t E\right) , \tag{4.1}$$

in the vicinity of the transition coupling. This guarantees that the integral in Eq. (3.4) can be evaluated accurately. We note that σ can be estimated with a standard MC simulation.

In practice we choose a value, $\delta = 0.005$, which clearly satisfies $\delta \ll \sigma$ (see Fig. 9 below) and do not undertake extensive investigations of the sensitivity to this choice.

C. Tuning $N_{\rm hit}$

Finally, we discuss the tuning of $N_{\rm hit}$, which we recall is the number of updates one does with a given $\omega_t(E)$ before updating to $\omega_{t+1}(E)$. To reduce computational effort, one wants to choose $N_{\rm hit}$ as small as possible. This, however, can introduce a sizable systematic error since the convergence of the WL algorithm is formally only guaranteed if $N_{\rm hit} \to \infty$. The lower limit $N_{\rm hit}$ depends on γ. This is because for large values of γ, the update of Eq. (3.6) is very abrupt and the system will require more hits to equilibrate into the new distribution $p_{t+1}(E)$. Correspondingly, for the very small values of γ that we use, the system may be able to equilibrate even with $N_{\rm hit} = 1$. In fact we use this value for most of our runs.

Lacking a firm theoretical foundation, it is clearly important to do numerical checks of the dependence on $N_{\rm hit}$. What we find (as will be shown below) is that $N_{\rm hit} = 1$ is acceptable (i.e. gives the same results as with larger values) if γ is small enough. A possible explanation

for this is that the number of effective hits between updates of the entropy is larger than $N_{\rm hit}$. This is because the typical change in E in an individual update is, in our simulations, an order of magnitude smaller than δ. Thus the system performs a random walk "inside" the Gaussian F_δ. So, in an approximate sense, the simulations are being done with effective values of $N_{\rm hit}$ and γ that are two orders of magnitude larger than the assigned values.

We conclude this section by stressing that this systematic error should be estimated explicitly. This can be done by comparing results for derived quantities to those obtained using standard MC simulations at values of b away from the transition (so that the latter are reliable), and/or by checking the sensitivity of the results obtained with WLR to changes in γ and $N_{\rm hit}$. Details of such checks will be described at the end of the next section.

V. RESULTS

We have undertaken long runs with $N = 20 - 50$ using the parameters listed in Tables I–IV. (The parameter we denote by $N_{\rm bin}$ is discussed in Appendix B.) In all cases the measurements were separated by 10000 full updates of the model (for a definition of a full update see Appendix A), except for the data in the last two rows of Table I, where the separation was by 100000 full updates. Thus for each choice of N and algorithm parameters, we perform in total $(1-3)\times 10^6$ full updates.

To present our results we first define the logarithm of the canonical probability function, calculated at the transition coupling b_t:

$$\overline{\omega}(E) = \omega(E) + 12\, N^2\, b_t(N)\, E. \tag{5.1}$$

We write $b_t(N)$ in Eq. (5.1) to emphasize that the transition coupling b_t depends on N. Presenting $\overline{\omega}(E)$ and not $\omega(E)$ makes the N dependence more apparent. We calculate $\omega(E)$ by averaging over the measurements. In Fig. 8 we present $\overline{\omega}(E)/N^2$ for the gauge groups we studied. The values of $b_t(N)$ used to generate this figure appear in Table VI and we discuss how we obtained them below. In Fig. 9 we show the canonical probability function itself, i.e. $\exp(\overline{\omega}(E))$. We do not show statistical errors in either figure since the meaning of such an error for both $\overline{\omega}(E)$ or its exponent is nontrivial: only differences of $\overline{\omega}(E)$ and ratios of $\exp(\overline{\omega}(E))$ have physical meaning. Meaningful errors can be computed using a simple

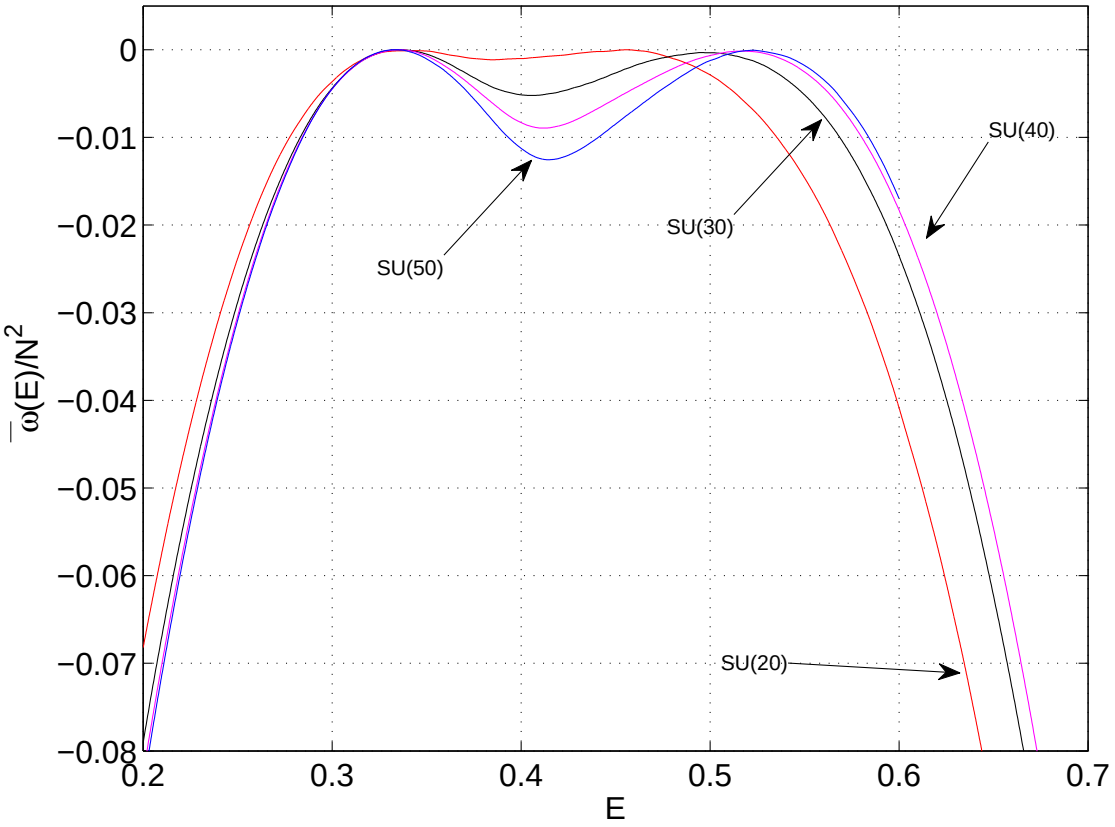

FIG. 8: Our most reliable estimates of $\overline{\omega}(E)/N^2$. All results were obtained with $N_{\rm hit} = 1$ and $\delta = 0.005$, except for $N = 50$ where $\delta = 0.0025$. The values of γ were $10^{-4}, 1.4\times10^{-5}, 2.5\times10^{-5}, 3\times10^{-5}$ for $N = 20, 30, 40, 50$ respectively. For presentation purposes we shift the maximum of $\overline{\omega}(E)$ to zero for each N. Note that a smaller $E_{\rm max}$ was used for $N = 50$.

scheme of error propagation but this is not necessary for our purposes here.[10] An indication of the size of the uncertainty is given, however, from the "wiggles" in Fig. 9. Note that these are much larger than those in Fig. 8, because of the exponential enhancement.

The expected double-peak structure is clearly seen, yet the WL algorithm has done its job by providing the density of states in the intermediate regime. It is noteworthy that the dip in $\overline{\omega}/N^2$ grows with increasing N. This is contrary to the usual behavior in field theories where the dip in this normalized quantity decreases as the $N_{\rm dof}$ increases.

[10] For example, one can estimate the statistical error in the ratio of $\exp(\overline{\omega}(E))$ between adjacent values of E and propagate it in a stochastic manner to find the error in $\exp(\overline{\omega}(E_1) - \overline{\omega}(E_2))$ for a finite difference $|E_1 - E_2|$.

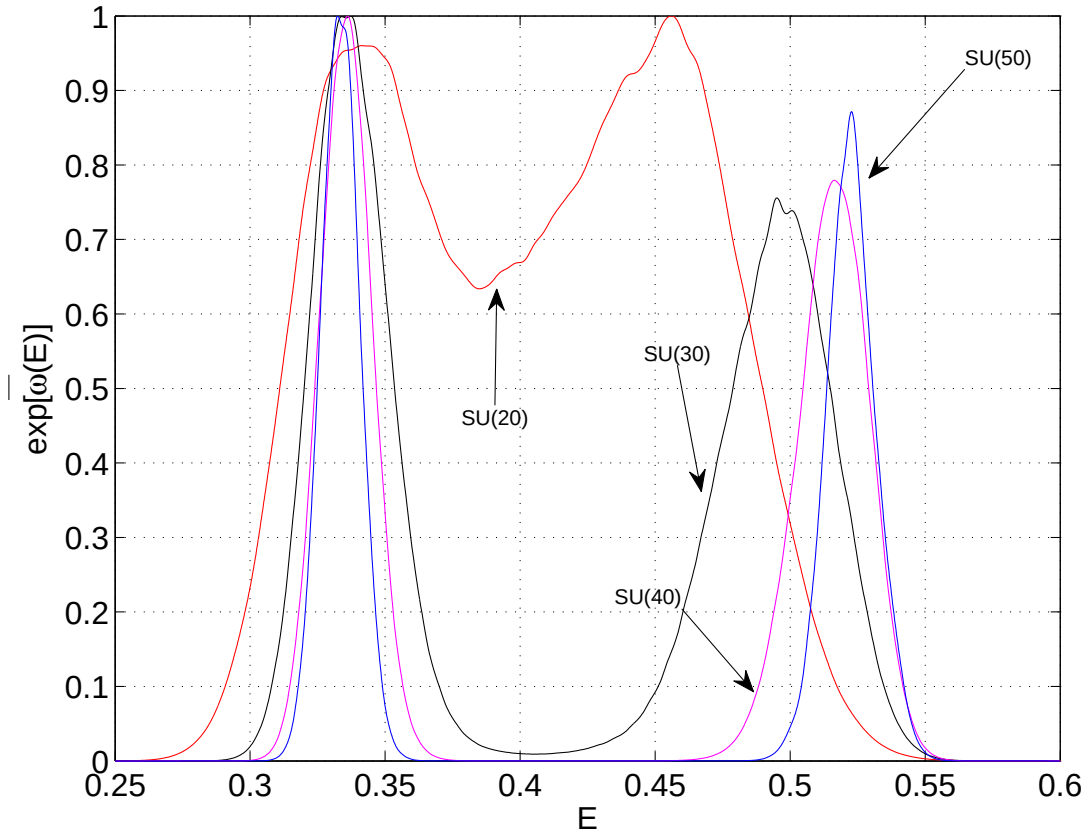

FIG. 9: Our most reliable estimates of $\exp(\overline{\omega}(E))$. The parameters are as in Fig. 8.

Using our estimates of $\omega(E)$ we can calculate the average action density $\mathcal{E}$ and the corresponding specific heat $\mathcal{C}$,

$$\mathcal{E}(b) = Z^{-1}(b) \int dE \, \exp\left[\omega(E) + 12N^2 bE\right] E, \tag{5.2}$$

$$\mathcal{C}(b) = Z^{-1}(b) \int dE \, \exp\left[\omega(E) + 12N^2 bE\right] (E - \mathcal{E}(b))^2 \, , \tag{5.3}$$

$$Z(b) = \int dE \, \exp\left[\omega(E) + 12N^2 bE\right] \, , \tag{5.4}$$

as a function of inverse 't Hooft coupling b, at least for the range of b where $\mathcal{E}$ lies well within our range $[E_{\min}, E_{\max}]$. The results of doing so are shown in Figs. 10 and 11. All statistical errors are obtained using the jackknife method, dropping single measurements of ω in turn from the average.

We define the transition coupling b_t to be the location of the peak in $\mathcal{C}(b)$ and give the resulting values of $b_t(N)$ in Tables I–IV. The WL algorithm is clearly able to determine b_t with high accuracy (0.1-0.2%). We note that error in b_t is roughly constant as N increases,

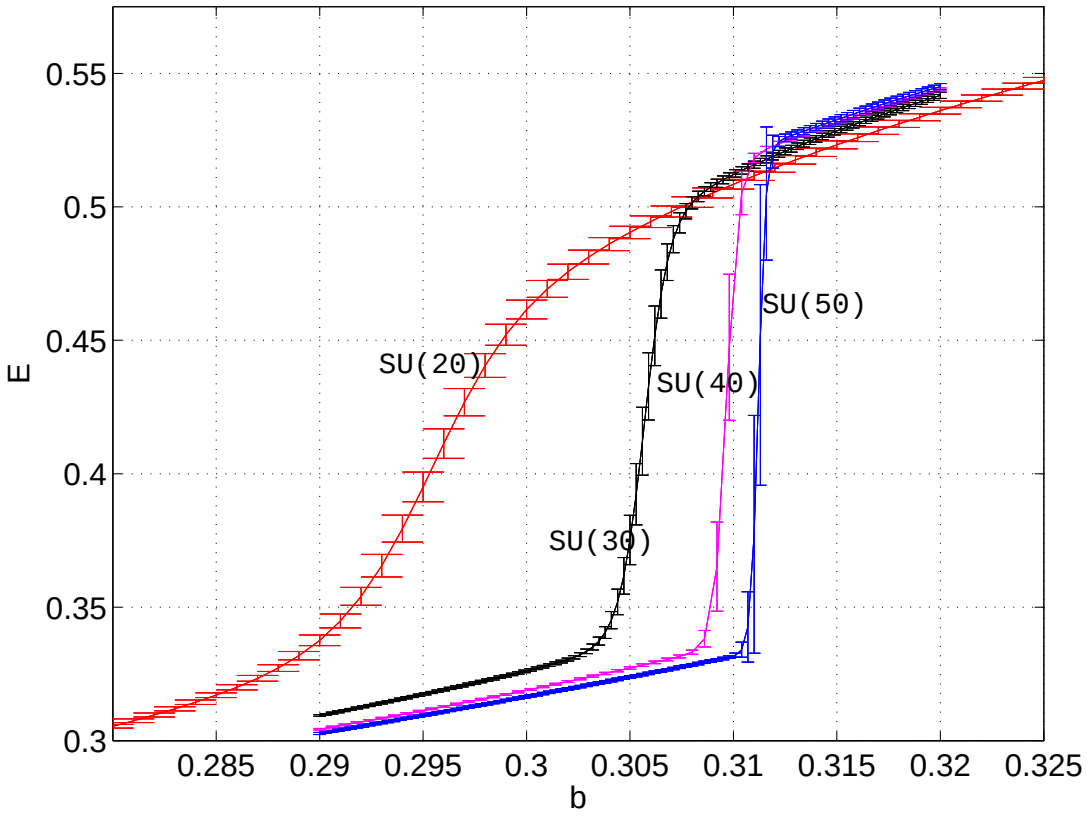

FIG. 10: Final results for $\mathcal{E}(b)$ obtained using WLR. Error bars are shown at selected values of b, and are highly correlated.

as long as we scale γ so that the tunneling rate stays approximately the same (as discussed in the previous section). Since the number of full updates is approximately the same for all N, this means that the computational effort is growing proportional to N^2. This is a much milder dependence than the exponential growth required for canonical simulations.

All results for b_t for a given N are consistent, despite the use of different values of the parameters of the algorithm. As a further check we have used the Ferrenberg-Swendsen multi-histogram reweighting method, which works well for $N = 20, 30$, but fails for $N \geq 40$. The values of b_t obtained with FSR are given in Table V, and agree within the very small errors with those from the WL algorithm.

The situation is different for the results for the peak value of the specific heat, $\mathcal{C}_t \equiv \mathcal{C}(b = b_t)$. Although results in Tab. I for $N = 20$ are consistent, those in Tabs. II and III for $N = 30$ and 40 are not. In addition, we find discrepancies with results from canonical MC

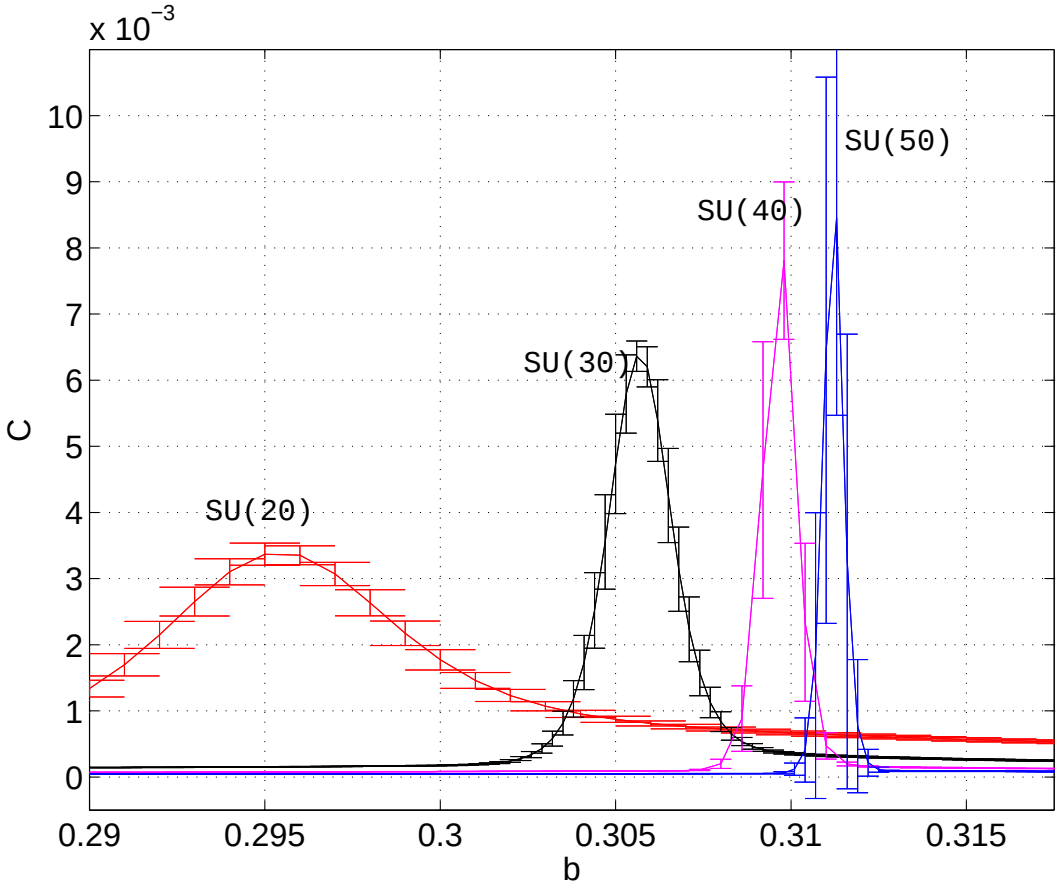

FIG. 11: Final results for $\mathcal{C}(b)$ obtained using WLR.

simulations. These are exemplified by Fig. 12, where we present the estimates of $\mathcal{E}(b \gtrsim 0.305)$ from WLR for $SU(30)$ together with the values obtained from direct MC simulations. What we find is that we obtain agreement with FSR and/or canonical MC results only if either γ is small enough or $N_{\rm hit}$ is large. This is presumably the realization of the systematic error discussed in Section IV and illustrates the importance of having results at more than one value of γ or $N_{\rm hit}$.

γ	$N_{\rm bin}$	$N_{\rm meas}$	total number of tunnelings	b_t	$\mathcal{C}_t$
10^{-4}	1000	108	~ 140	0.29585(45)	$3.222(128) \times 10^{-3}$
10^{-4}	1000*	110	~ 400	0.29597(21)	$3.336(55) \times 10^{-3}$
10^{-4}	4000	100	~ 120	0.29544(37)	$3.399(55) \times 10^{-3}$
10^{-5}	1000	18	~ 80	0.29509(37)	$3.260(83) \times 10^{-3}$
10^{-6}	1000	20	~ 40	0.29585(13)	$3.402(35) \times 10^{-3}$

TABLE I: Results using WLR for $SU(20)$, using the range $E \in [0.1, 0.7]$, and setting $\delta = 0.005$ and $N_{\rm hit} = 1$. The row denoted by a star was obtained by adding explicit permutations to the update of the $SU(N)$ matrices, which were found to be accepted 20% of the time. For further details on the importance of permutations we refer to Ref. [12].

γ	$N_{\rm hit}$	$N_{\rm bin}$	$N_{\rm meas}$	total number of tunnelings	b_t	$\mathcal{C}_t$
2.25×10^{-4}	1	1500	108	~ 140	0.30557(55)	$5.974(50) \times 10^{-3}$
2.25×10^{-4}	1	50000	100	~ 180	0.30652(80)	$5.051(260) \times 10^{-3}$
2.25×10^{-4}	1740	1500	114	~ 20	0.30539(35)	$6.276(120) \times 10^{-3}$
1.4×10^{-5}	1	1500	102	~ 60	0.30569(17)	$6.395(230) \times 10^{-3}$
1.4×10^{-4}	1	1500	110	~ 20	0.30551(30)	$6.453(120) \times 10^{-3}$

TABLE II: Results from WLR for $SU(30)$, using the range $E \in [0.1, 0.7]$. All calculations were done with $\delta = 0.005$ except for that presented in the last row, for which $\delta = 0.0008$.

VI. SUMMARY

In this paper we present an implementation of a variant of the Wang-Landau reweighting algorithm in the context of $SU(N)$ lattice gauge theories.[11]

This algorithm was introduced in the field of statistical mechanics to calculate the density of states of discrete spin systems. We use a generalization of the original algorithm to systems with continuous degrees of freedom and apply it to a matrix model that is obtained by quenched reduction from four-dimensional $SU(N)$ lattice gauge theory. This matrix model

[11] The original WL algorithm was implimented for $U(1)$ gauge theory in Ref. [5], and used to provide an input weighting function for a multicanonical simulation.

γ	$N_{\rm meas}$	total number of tunnelings	b_t	$\mathcal{C}_t$
4×10^{-4}	151	~ 850	0.30965(75)	$6.67(30) \times 10^{-3}$
2.5×10^{-5}	355	~ 75	0.30968(20)	$8.24(10) \times 10^{-3}$

TABLE III: Results from WLR for $SU(40)$, using the range $E \in [0.1, 0.7]$, and obtained with $\delta = 0.005$, $N_{\rm bin} = 1875$, and $N_{\rm hit} = 1$.

γ	$N_{\rm meas}$	total number of tunnelings	b_t	$\mathcal{C}_t$
3×10^{-5}	108	~ 45	0.31119(19)	$9.02(12) \times 10^{-3}$

TABLE IV: Results from the WLR for $SU(50)$, using the range $E \in [0.1, 0.6]$, and obtained with $\delta = 0.0025$, $N_{\rm bin} = 2100$, and $N_{\rm hit} = 1$.

consists of four $SU(N)$ matrices with interactions governed by the 't Hooft coupling λ, and has a first-order strong-to-weak coupling phase transition in its large-N limit at $\lambda = \lambda_t$. An accurate measurement of λ_t at $N = \infty$ is what we aimed to achieve using WLR.

Our variant of the WL algorithm does not extrapolate the flucturations in the Boltzman weights towards zero, but rather retains these fluctuations at a small, non-zero value, in order to maintain tunneling at a first-order transition. Assuming these fluctuations are symmetric around zero, we can systematically estimate the error in the density of states and in derived quantities such as the specific heat. We have studied the systematic errors associated with chosing the various parameters of the algorithm. Our most reliable WL estimates of λ_t for gauge groups with $N = 20, 30, 40, 50$ are summarized in Table VI and plotted in Fig. 13 versus $1/N^2$. We fit our data to the form

N	Reweighting using	b_t	$\mathcal{C}_t$
20	40 histograms	0.295980(48)	$3.32(3) \times 10^{-3}$
30	17 histograms	0.30551(30)	$6.45(5) \times 10^{-3}$

TABLE V: Results from FSR multi-histogram reweighting. For $N = 20(30)$ each histogram contains an average of 10^4 (5×10^4) measurements, separated by 5 full model updates from each other.

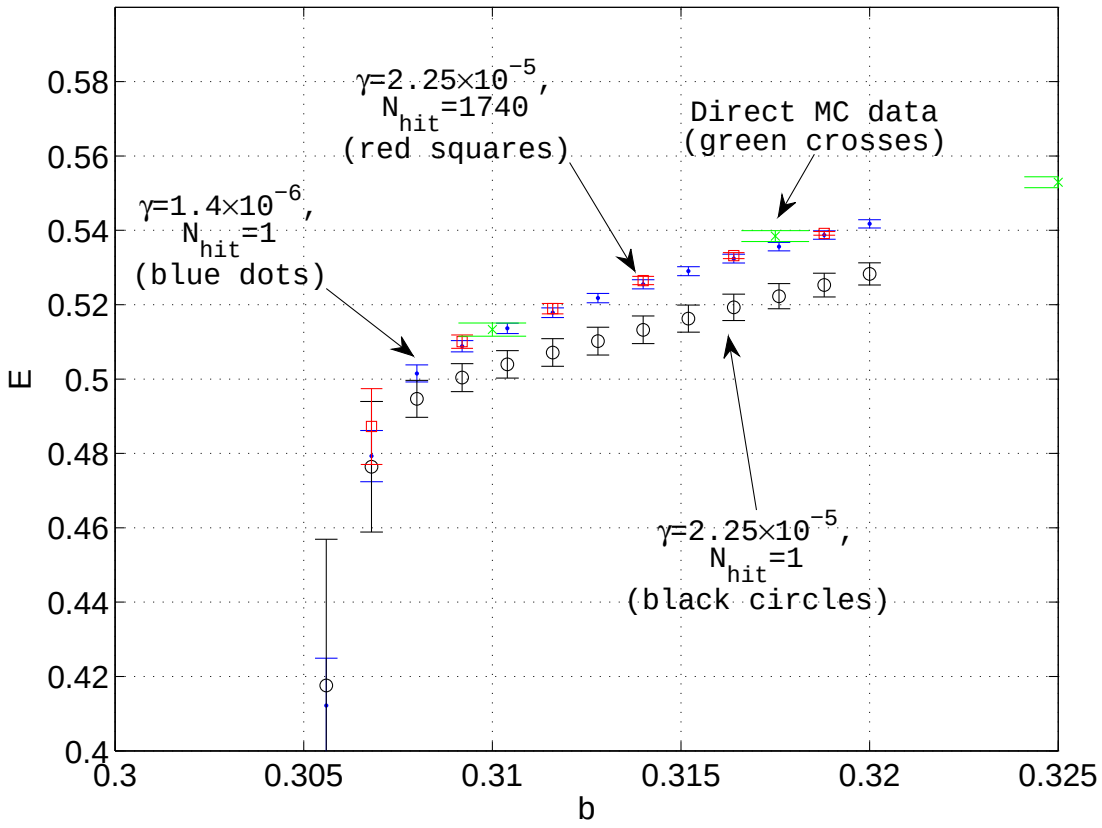

FIG. 12: Comparison of WLR results for the average action density for $SU(30)$, with direct measurements from standard MC simulations. The systematic error discussed in Section III C is clearly seen.

N	20	30	40	50
$b_t(N) = (\lambda_t)_N^{-1}$	0.29544(37)	0.30569(17)	0.30968(20)	0.31121(19)

TABLE VI: A summary of our most reliable WL results for $b_t(N) = (\lambda_t)_N^{-1}$.

$$(\lambda_t)_N^{-1} = (\lambda_t)_\infty^{-1} + \frac{A}{N} + \frac{B}{N^2}, \tag{6.1}$$

and present the results of these fits in Table VII. We also plot the result of the linear fit in $1/N^2$ (i.e. the fit with $A = 0$ whose results are presented in the first row of Table VII) in Fig. 13.

We find that in the large-N limit $(\lambda_t)_\infty^{-1} = 0.3142(2) - 0.3148(10)$, depending on the way we fit. These results are many standard deviations away from the value of $(\lambda)_{\rm Bulk}^{-1} \simeq 0.36$

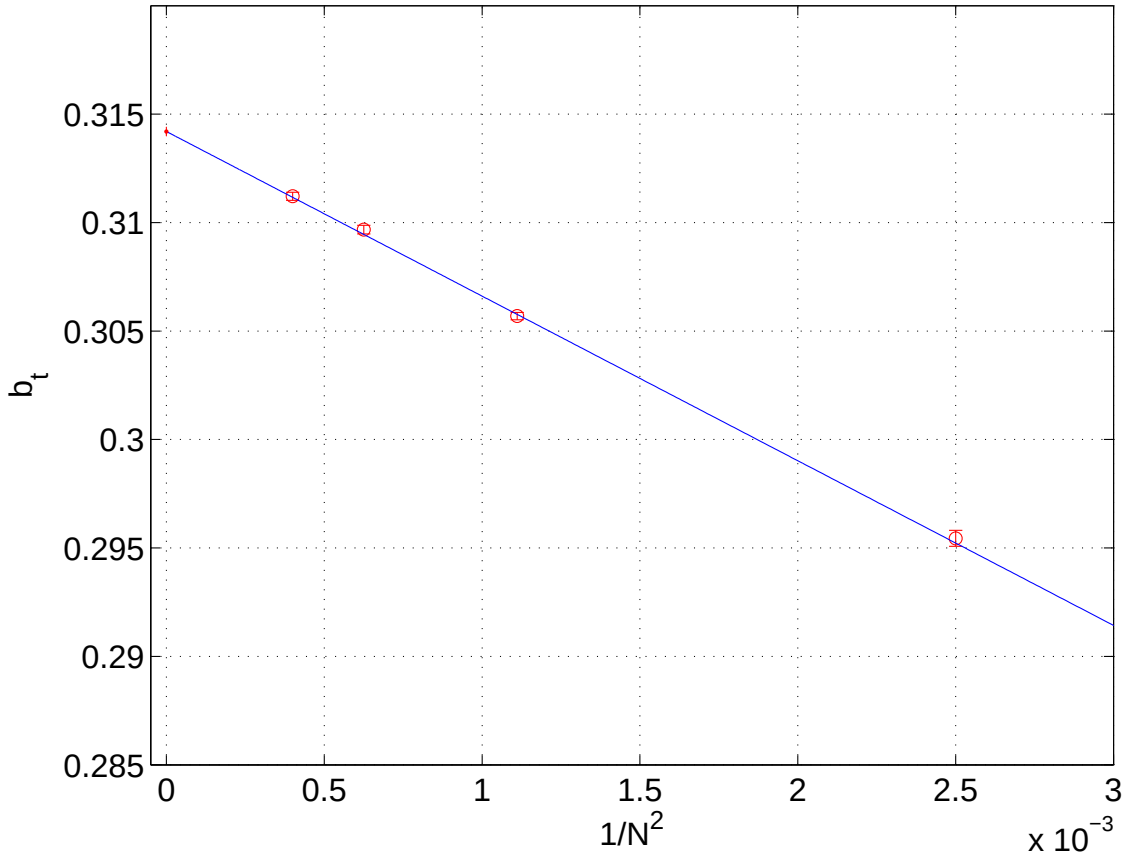

FIG. 13: The strong-to-weak transition coupling, $b_t = 1/\lambda_t$, plotted versus $1/N^2$. [Red] squares show our results using the Wang-Landau algorithm from Table VI. The solid blue curve is the linear fit described in the text (with parameters listed in the first row of Table VII).

Type of fit	$(\lambda_t)^{-1}_\infty$	A	B	χ^2/d.o.f.
$A = 0, B \neq 0$	0.3142(2)	–	-7.59(18)	1.45/2
$A \neq 0, B \neq 0$	0.3148(10)	-0.037(65)	-7.06(97)	1.1/1

TABLE VII: The fit parameters $(\lambda_t)^{-1}_\infty$, A, and B, obtained from fitting the Wang-Landau data in Table VI to the form Eq. (6.1). The first row shows results from a fit linear in $1/N^2$ (i.e. with $A = 0$).

where the strongly first order 'bulk' transition takes place in four-dimensional $SU(\infty)$ lattice gauge theories. This discrepancy is one of several pieces of evidence adduced in Ref. [12] for the breakdown of large-N quenched reduction in four-dimensional $SU(N)$ lattice gauge theories, and we refer the reader to that paper for further discussion. Such a discrepancy

was not seen in past explorations of the matrix model partly because the phase transition is so strong that it is very hard to measure its transition coupling by conventional means. The Wang-Landau algorithm allowed us to solve this problem and to determine that there is a discrepancy. We conclude that the Wang-Landau algorithm can be a useful and feasible way to study $SU(N)$ lattice gauge theories.

Acknowledgments

We acknowledge A. Bazavov, B. Berg and S. Trebst for useful discussions. This work was supported in part by the U.S. Department of Energy under Grant No. DE-FG02-96ER40956.

APPENDIX A: ALGORITHMS FOR SIMULATING QUENCHED, REDUCED $SU(N)$ LATTICE GAUGE THEORY

In our standard (non Wang-Landau) MC simulations of the model we used several different algorithms to update the matrices V_μ. We use a standard Metropolis algorithm (M), a "hybrid" heat-bath + Metropolis (HM), and a full heat-bath (HB)—the latter including different types of over-relaxations. For the M and HM algorithms we generate a set of random $SU(2)$ matrices at the beginning of the run and keep them in memory. For each matrix in this list we add to the list its inverse. The M algorithm is completely standard. We randomly choose an $SU(2)$ matrix u from the list, extend it to an $SU(N)$ matrix by adding 1's along the diagonal, update $V_\mu \to u\, V_\mu$ and accept this proposed update with the usual Metropolis probability. This is repeated five times for equilibration. This process is then repeated, following Cabibbo and Marinari [19], for each of the $[N(N-1)/2]$ $SU(2)$ subgroups of $SU(N)$ in turn, and for each V_μ in turn.

We now describe the other algorithms, which are less standard.

1. Hybrid Heat-bath Algorithm

Here we use the prescription suggested in Ref. [20] to make the action linear in the link matrices U_μ. This requires a Hubbard-Stratonovich Gaussian field $Q_{\mu\nu}$ for each plaquette. It results in an effective action $A_{\rm eff}(U_\mu, Q_{\mu\nu})$ that is quadratic in $Q_{\mu\nu}$ and linear in U_μ—and

thus quadratic in V_μ. We then update one of the V_μ as follows:

1. First update the matrices $Q_{\mu,\nu}$. This update is trivial since Q has a (shifted) Gaussian distribution.

2. Update V_μ as in the M algorithm using all $SU(2)$ subgroups but now with the action $A_{\rm eff}$.

This is repeated in turn for each of the links.

2. Heatbath and over-relaxation algorithms

The heatbath algorithm requires the use of two auxiliary fields in order to obtain an action which is linear in the V_μ. It is not quite as simple as applying the approach of Ref. [20] twice, and so we give some details.

We begin by recalling the action

$$A = 2N \sum_{\mu<\nu} \mathrm{Re\,Tr}\, \left(U_\mu U_\nu U_\mu^\dagger U_\nu^\dagger \right) , \tag{A1}$$

where $U_\mu = V_\mu \Lambda_\mu V_\mu^\dagger$ and $(\Lambda_\mu)_{ab} = \delta_{ab} \exp(ip_\mu^a)$. We next define two sets of unitary matrices

$$A_{\mu\nu} \equiv V_\mu^\dagger V_\nu = A_{\nu\mu}^\dagger , \qquad B_{\mu\nu} \equiv A_{\mu\nu} \Lambda_\nu A_{\mu\nu}^\dagger \neq B_{\nu\mu}^\dagger , \qquad (\mu \neq \nu) , \tag{A2}$$

in terms of which the action can be written as

$$\begin{aligned} A &= 2N \sum_{\mu<\nu} \mathrm{Re\,Tr}\, \left(A_{\mu\nu} \Lambda_\nu A_{\mu\nu}^\dagger \Lambda_\mu^\dagger A_{\mu\nu} \Lambda_\nu^\dagger A_{\mu\nu}^\dagger \Lambda_\mu \right) , & \text{(A3)} \\ &= 2N \sum_{\mu<\nu} \mathrm{Re\,Tr}\, \left(B_{\mu\nu} \Lambda_\mu^\dagger B_{\mu\nu}^\dagger \Lambda_\mu \right) . & \text{(A4)} \end{aligned}$$

For each plaquette, i.e. for each $\mu < \nu$ we introduce auxiliary complex fields $\widetilde{Q}_{\mu\nu}$ and $\widetilde{P}_{\mu\nu}$, with Boltzmann weights

$$\exp[-bN\, \mathrm{Tr}\, (\widetilde{Q}_{\mu\nu} \widetilde{Q}_{\mu\nu}^\dagger) - bN\, \mathrm{Tr}\, (\widetilde{P}_{\mu\nu} \widetilde{P}_{\mu\nu}^\dagger)] . \tag{A5}$$

These are then shifted as follows:

$$\begin{aligned} Q_{\mu\nu} &= \widetilde{Q}_{\mu\nu} + \left\{ B_{\mu\nu}, \Lambda_\mu^\dagger \right\} , & \text{(A6)} \\ Q_{\mu\nu}^\Lambda &= \{ Q_{\mu\nu}, \Lambda_\mu \} = \left\{ \widetilde{Q}_{\mu\nu}, \Lambda_\mu \right\} + 2B_{\mu\nu} + \Lambda_\mu B_{\mu\nu} \Lambda_\mu^\dagger + \Lambda_\mu^\dagger B_{\mu\nu} \Lambda_\mu , & \text{(A7)} \\ P_{\mu\nu} &= \widetilde{P}_{\mu\nu} + A_{\mu\nu} \Lambda_\nu^\dagger + {Q_{\mu\nu}^\Lambda}^\dagger A_{\mu\nu} . & \text{(A8)} \end{aligned}$$

The staple-like quantity $X_{\mu\nu}$ can then be calculated:

$$X_{\mu\nu} = \Lambda^\dagger_\nu P^\dagger_{\mu\nu} + P^\dagger_{\mu\nu} {Q^\Lambda_{\mu\nu}}^\dagger\,, \qquad X_{\nu\mu} = X^\dagger_{\mu\nu}\,, \qquad (\mu < \nu) \tag{A9}$$

Finally, the action can be written in a form suitable for a heatbath or overrelaxed update:

$$\begin{aligned} A' &= A - N \sum_{\mu<\nu} \Big[\mathrm{Tr}\,(\widetilde{Q}_{\mu\nu}\widetilde{Q}^\dagger_{\mu\nu}) + \mathrm{Tr}\,(\widetilde{P}_{\mu\nu}\widetilde{P}^\dagger_{\mu\nu}) + \mathrm{const.} \Big] && \text{(A10)} \\ &= N \sum_{\mu\neq\nu} \mathrm{Tr}\,\left(V_\mu X_{\nu\mu} V^\dagger_\nu \right) \\ &\quad -N \sum_{\mu<\nu} \Big[\mathrm{Tr}\,(Q_{\mu\nu}Q^\dagger_{\mu\nu}) + \mathrm{Tr}\,(Q^\Lambda_{\mu\nu}{Q^\Lambda_{\mu\nu}}^\dagger) + \mathrm{Tr}\,(P_{\mu\nu}P^\dagger_{\mu\nu}) \Big]\,. && \text{(A11)} \end{aligned}$$

In summary, to evaluate an observable that depends only on the gauge fields one can use

$$\begin{aligned} \langle O(U) \rangle &= \frac{1}{Z} \int \prod_\mu DV_\mu\, e^{b\,A}\, O(U) && \text{(A12)} \\ &= \frac{1}{Z'} \int \prod_\mu DV_\mu \prod_{\mu<\nu} D\widetilde{Q}_{\mu\nu} D\widetilde{Q}^\dagger_{\mu\nu} D\widetilde{P}_{\mu\nu} D\widetilde{P}^\dagger_{\mu\nu}\, e^{b\,A'}\, O(U) && \text{(A13)} \\ &= \frac{1}{Z'} \int \prod_\mu DV_\mu \prod_{\mu<\nu} DQ_{\mu\nu} DQ^\dagger_{\mu\nu} DP_{\mu\nu} DP^\dagger_{\mu\nu}\, e^{b\,A'}\, O(U)\,, && \text{(A14)} \end{aligned}$$

where

$$Z' = \int \prod_\mu DV_\mu \prod_{\mu<\nu} D\widetilde{Q}_{\mu\nu} D\widetilde{Q}^\dagger_{\mu\nu} D\widetilde{P}_{\mu\nu} D\widetilde{P}^\dagger_{\mu\nu}\, e^{b\,A'}\,. \tag{A15}$$

The form (A14), together with the expression for A' in eq. (A11), shows how the auxiliary fields decouple the V's. To get the correctly distributed V's and P's, one can update V_μ using the "staple" part of the action:

$$A_{\mathrm{staple}}(V_\mu) = 2N \sum_{\nu\neq\mu} \mathrm{Re}\,\mathrm{Tr}\,(V_\mu X_{\nu\mu} V^\dagger_\nu)\,, \tag{A16}$$

where we stress that in this case the sum is now only over ν. This form is suitable for a heat-bath update, which we implement in each $SU(2)$ subgroup in turn.

To generate the correct distribution of the P's is straightforward. Given the Q's and V's, one can generate $\widetilde{P}$'s with the Gaussian measure and make the shifts given in eq. (A8). This leads to the correct linear and quadratic terms in $P_{\mu\nu}$ in the action of eq. (A11).

To update the Q's one must be more careful. Simply generating $\widetilde{Q}$'s with Gaussian measure and using the shift of eq. (A6) leads to the *wrong* distribution: neither the quadratic or the linear terms in eq. (A11) are reproduced. Instead, one should "complete the square"

using the terms that are present in eq. (A11). To do so requires that one first generate the $\widetilde{Q}$'s using the Gaussian measure, but then shifts and rescales as follows:

$$(Q_{\mu\nu})_{ab} = \alpha_{ab}(\widetilde{Q}_{\mu\nu})_{ab} + \alpha_{ab}^2(\{A_{\mu\nu}P_{\mu\nu}^\dagger, \Lambda_\mu^\dagger\})_{ab}\,, \tag{A17}$$

$$\alpha_{ab} = \frac{1}{\sqrt{3 + 2\cos(p_\mu^a - p_\mu^b)}}\,. \tag{A18}$$

Thus the structure of the algorithm is as follows. One begins with an initial choice of V's *and* Q's. Then one can update all the P's, update all the Q's, and finally update the V's (updating all directions for given Q's and P's). To return to the beginning of the loop one needs to store not only the V's but also the Q's. One could also interchange the ordering and roles of the P's and Q's.

Once one has an effective action that is linear in the $U(N)$ matrices V_μ the way is open for over-relaxation algorithms. We have thus implemented both an over-relaxation in all the $SU(2)$ subgroups of $SU(N)$ as well as a full $SU(N)$ over-relaxation of the type described in [21].

3. Update scheme in the Wang-Landau algorithm

In the WL algorithm we need to propose changes to the V_μ. This we do as in the Metropolis and HM algorithms, i.e. one $SU(2)$ subgroup at a time. Such an update is what we refer to as a "hit", so if $N_{\rm hit} = 1$ we change $\omega_t(E)$ after each individual $SU(2)$ multiplication. We also checked that updating $\omega(E)$ in between full $SU(N)$ updates (for all four V_μ) gives similar results—this gives $N_{\rm hit} = 2 \times N(N-1)$. In either case, we call a "full update" the update of all $SU(2)$ subgroups for all four links.

Finally, we have considered an extra type of update which permutes the angles $p_\mu^a \leftrightarrow p_\mu^b$ for randomly chosen pairs of indices a and b. This was motivated by the importance of permutations in this quenched-reduced model [12].

APPENDIX B: MORE PRACTICAL ISSUES

In this section we list several practical issues relevant to the implementation of WLR.

1. A initial guess for $\omega_{t=0}(E)$

We suggest performing the first implementation of the WLR with a relatively low value of $N_{\rm dof} = N_{\rm dof}^{(0)}$. This run can begin with a 'blind' initial guess of $\omega_0(E; N_{\rm dof}^{(0)}) = 0$. We then found it useful to appropriately scale the best estimate of $\omega(E, N_{\rm dof}^{(0)})$ with $N_{\rm dof}$, so as to use it as a good initial guess for $N_{\rm dof} > N_{\rm dof}^{(0)}$. Since $\omega(E)$ is an extensive quantity this means setting

$$\omega_{t=0}(E; N_{\rm dof}^{(1)}) = \frac{N_{\rm dof}^{(1)}}{N_{\rm dof}^{(2)}} \omega_{t=\infty}(E; N_{\rm dof}^{(2)}). \tag{B1}$$

In our case, with $N_{\rm dof} \sim N^2$, this corresponds to $\omega_{t=0}(E; N_1^2) = \left(\frac{N_1}{N_0}\right)^2 \omega_{t=\infty}(E; N_0^2)$. The generalization for a field theory in a finite lattice volume is obvious. We found that this procedure shortens the initial stage of WLR considerably.

2. Boundary effects

As we mention in Section IV it is useful to use WLR in a subset of the full range of E. This is sufficient if at the values of b of interest, the average action density $\mathcal{E}$ is localized far from the regime's boundaries.

This modification complicates the theoretical analysis of Sec. III B in a way we only partially addressed.

The presence of the boundaries raises two practical questions

1. What do we do when the update $E_{\rm old} \to E_{\rm new}$ results in a value $E_{\rm new}$ which is outside of the region $[E_{\rm min}, E_{\rm max}]$?

2. What do we do when we update $\omega_t \to \omega_{t+1}$ and the update function $F_\delta(E, E_t)$ extends outside the desired region?

In this work we generalize the proposals of Ref. [22]. The answer to the first question is that we reject $E_{\rm new}$ and thus perforce stay inside the desired region. This means setting $E_{\rm new} = E_{\rm old}$ and updating $\omega_t(E_{\rm new})$ as in a regular update. This is the standard approach, which one can understand as follows. If one were simulating the full range of E then every time one left the range $[E_{\rm min}, E_{\rm max}]$ one would eventually return. We are just dropping the MC-time history of the WL algorithm for which E was outside the desired range. Since at

time t the WL algorithm updates $\omega_t(E)$ only around E_t, performing WLR in this way gives an estimate for $\omega(E)$ which is correct away from the boundaries $E = E_{\rm min,max}$.

Our answer to the second question is to "reflect" the part of $F_\delta(E, E_t)$ which lies outside of the desired range back into the range. The precise definition of the reflection is as follows. For all E (even those outside $[E_{\rm min}, E_{\rm max}]$) perform

$$\omega(E') \to \omega(E') + \gamma F(E, E_{\rm old})\,, \tag{B2}$$

with

$$E' = \begin{cases} E & \text{for} \quad E \in [E_{\rm min}, E_{\rm max}], \\ 2E_{\rm min} - E & \text{for} \quad E < E_{\rm min}, \\ 2E_{\rm max} - E & \text{for} \quad E > E_{\rm max}. \end{cases} \tag{B3}$$

Here we assume that E' always obeys $E' \in [E_{\rm min}, E_{\rm max}]$. This is valid as long as the interval size $(E_{\rm max} - E_{\rm min})$ is larger than the average change $|E_{\rm new} - E_{\rm old}|$, which is very well satisfied in practice.[12]

The reflected update has two important properties. First, it maintains the result that the area $\int dE\, F_\delta(E, E_t)$ is independent of E_t. Second, for our Gaussian choice of F_δ, the definition remains symmetric: $F_\delta(E, E_t) = F_\delta(E_t, E)$.

We find that if one does not perform the updates that corresponds to last two rows in Eq. (B3), then the WLR fails to converge, and effectively overestimates $\omega(E)$ near the boundaries. This is easy to understand: if we do not reflect the contribution of the Gaussian in Eq. (B2) then effectively the update to $\omega(E)$ close to the boundary is smaller than it would be in the absence of the boundary.[13] Thus the force that drives one away from the boundary is too weak and one spends too much time updating $\omega_t(E)$ near the boundary.

3. Storing the functions $\omega_t(E)$ and $h_t(E)$

Despite the continuous fashion in which we implement the WL algorithm, one still needs to store the functions $\omega_t(E)$ and $h_t(E)$ in memory, as well as to update them. We do this

[12] We note in passing that once F_δ drops below a certain size (10^{-8} was our choice) we set it to zero. This saves calculation time since one needs to update $\omega(E)$ only for a subset of E, and also avoids the problem of double and higher-order reflections.

[13] To demonstrate this one can take the limit of $\delta \to 0$ and reproduce the case discussed in [22]. We also note that maintaining the area under the F_δ plays an important role in the theoretical analysis of Sec. III B.

by dividing the range $[E_{\rm min}, E_{\rm max}]$ into $N_{\rm bin}$ bins. The criterion that determines the bin size $\delta E = (E_{\rm max} - E_{\rm min})/N_{\rm bin}$ is the same as that for δ (see Sec. IV B). In order for the error in the numerical evaluation of the integral Eq. (3.4) by binning remain small, one must have

$$\delta E \ll \sigma, , \tag{B4}$$

where σ is the width in E of the canonical distribution function $P_C(E)$ of Eq. (4.1). Assuming that $\bar{\omega}(E)/N^2$ is quadratic about the peak and has a good $N \to \infty$ limit (which appears to hold for the "outside" branches of the peaks—see Fig. 8), then one can show that $\sigma \propto 1/N$. Thus one must increase the number of bins as $N_{\rm bin} \propto N$, as we have done (see Tables I-IV).

Another criterion one might consider using is to enforce a relationship between δE and the average step size. Naively one might think that the bins should be smaller than the average step size, so that discretization effects do not hinder the motion in E-space. This would be an onerous requirement, since our step size, which is $\sim (1-5) \times 10^{-4}$, would require significantly more bins than we use in most simulations. In fact, it turns out that the acceptance rate, and thus the motion through E-space is almost independent of the bin size. We have seen this numerically for $SU(30)$, where we have one simulation with 50000 bins. But one can also understand this analytically if the step size is much smaller than σ, which is the case for our simulations. We do not present the derivation, but the essential point is that with bins much larger than the step size the smaller acceptance when jumping between bins (when the step is to lower E) is exactly counterbalanced by the free motion (without rejection) within the bins.

---

[1] A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **61** (1988) 2635. Phys. Rev. Lett. **63**, 1195 (1989).

[2] B. A. Berg, "Introduction to Markov Chain Monte Carlo Simulations and their Statistical Analysis," *World Scientific* 2004. arXiv:cond-mat/0410490.

[3] B. A. Berg and T. Neuhaus, Phys. Rev. Lett. **68** (1992) 9 [arXiv:hep-lat/9202004].

[4] M. Campostrini, Nucl. Phys. Proc. Suppl. **73**, 724 (1999) [arXiv:hep-lat/9809072].

[5] B. A. Berg and A. Bazavov, Phys. Rev. D **74**, 094502 (2006) [arXiv:hep-lat/0605019].

[6] F. Wang and D. P. Landau, Phys. Rev. Lett. **86** (2001) 2050.

[7] G. Bhanot, K. Bitar and R. Salvador, Phys. Lett. B **188**, 246 (1987).

[8] Z. Fodor, S. D. Katz and C. Schmidt, JHEP **0703**, 121 (2007) [arXiv:hep-lat/0701022].

[9] A. Denbleyker, D. Du, Y. Liu, Y. Meurice and A. Velytsky, arXiv:0807.0185 [hep-lat].

[10] Chenggang Zhou, R. N. Bhatt, Phys. Rev. E **72**, 025701(R) (2005), arXiv:cond-mat/0306711.

[11] G. Bhanot, U. M. Heller and H. Neuberger, Phys. Lett. B **115**, 237 (1982). A. A. Migdal, Phys. Lett. B **116**, 425 (1982). D. J. Gross and Y. Kitazawa, Nucl. Phys. B **206**, 440 (1982). G. Parisi, Phys. Lett. B **112**, 463 (1982). G. Parisi and Y. C. Zhang, Nucl. Phys. B **216**, 408 (1983). G. Parisi and Y. C. Zhang, Phys. Lett. B **114** (1982) 319.

[12] B. Bringoltz and S. R. Sharpe, arXiv:0805.2146 [hep-lat].

[13] S. Trebst, D. A. Huse, and M. Troyer, Phys. Rev. E **70**, 046701 (2004).

[14] P. Dayal, S. Trebst, S. Wessel, D. Würt, M. Troyer, S. Sabhapandit, and S. N. Coppersmith, Phys. Rev. Lett. **92**, 097201 (2004).

[15] M. Okawa, Phys. Rev. Lett. **49**, 705 (1982).

[16] G. Brown and T. C. Schulthess, J. Appl. Phys. **97**, 10E303 (2005), S. Sinha and S. K. Roy, arXiv:0711.1031

[17] A. Laio and M. Parrinello, Proc. Nat. Acad. Sci. U.S.A. **99**, 12562 (2002), www.pnas.org/cgi/doi/10.1073/pnas.202427399. Y. Wu, J. D. Schmitt, and R. Car, J. Chem. Phys. **121**, 1193 (2004).

[18] Chenggang Zhou, T. C. Schulthess, Stefan Torbrgge, D. P. Landau, Phys. Rev. Lett. **96**, 120201 (2006), arXiv:cond-mat/0509335.

[19] N. Cabibbo and E. Marinari, Phys. Lett. B **119** (1982) 387.

[20] K. Fabricius and O. Haan, Phys. Lett. B **143**, 459 (1984).

[21] M. Creutz, Phys. Rev. D **36**, 515 (1987). P. de Forcrand and O. Jahn, Nucl. Phys. Proc. Suppl. **129**, 709 (2004) [arXiv:hep-lat/0309153]. J. Kiskis, R. Narayanan and H. Neuberger, Phys. Lett. B **574**, 65 (2003) [arXiv:hep-lat/0308033]. P. de Forcrand and O. Jahn, arXiv:hep-lat/0503041.

[22] B. J. Schulz, K. Binder, M. Müller, and D. P. Landau, Phys. Rev. E. **67**, 067102 (2003), arXiv:cond-mat/0302123.